Design Vision™
User Guide

Version C-2009.06, June 2009

SYNOPSYS®

Copyright Notice and Proprietary Information

Right to Copy Documentation

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Trademarks (™)

Service Marks (^{SM})

Contents

3. Performing Basic Tasks

4. Solving Timing Problems

Preface

This preface includes the following sections:

- What's New in This Release
- About This User Guide
- Customer Support

What's New in This Release

Information about new features, enhancements, and changes, along with known problems and limitations and resolved Synopsys Technical Action Requests (STARs), is available in the *Design Vision Release Notes* in SolvNet.

To see the *Design Vision Release Notes*,

1. Go to the Download Center on SolvNet located at the following address:

   https://solvnet.synopsys.com/DownloadCenter

   If prompted, enter your user name and password. If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.
2. Select Design Vision, and then select a release in the list that appears.

About This User Guide

This section contains information about the target audience of this document, where to find other pertinent publications, and documentation conventions used in this manual.

Audience

This user guide is for logic design engineers who have some experience using Design Compiler and who want to use the visualization features of Design Vision for synthesis or analysis. To use this user guide, you should be familiar with

- Synthesis using Design Compiler
- VHDL or Verilog HDL
- The UNIX or Linux operating system

Related Publications

For additional information about Design Vision, see Documentation on the Web, which is available through SolvNet at the following address:

https://solvnet.synopsys.com/DocsOnWeb

You might also want to refer to the documentation for the following related Synopsys products:

- Design Compiler
- DFT Compiler
- Power Compiler

Conventions

The following conventions are used in Synopsys documentation.

Convention	Description
`Courier`	Indicates command syntax.
`Courier italic`	Indicates a user-defined value in Synopsys syntax, such as *`object_name`*. (A user-defined value that is not Synopsys syntax, such as a user-defined value in a Verilog or VHDL statement, is indicated by regular text font italic.)
`Courier bold`	Indicates user input—text you type verbatim—in Synopsys syntax and examples. (User input that is not Synopsys syntax, such as a user name or password you enter in a GUI, is indicated by regular text font bold.)
[ ]	Denotes optional parameters, such as *`pin1`* `[`*`pin2 ... pinN`*`]`
|	Indicates a choice among alternatives, such as `low | medium | high` (This example indicates that you can enter one of three possible values for an option: low, medium, or high.)
_	Connects terms that are read as a single term by the system, such as `set_annotated_delay`
Control-c	Indicates a keyboard combination, such as holding down the Control key and pressing c.
\	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Edit > Copy	Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.

Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Support Center.

Accessing SolvNet

SolvNet includes an electronic knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. SolvNet also gives you access to a wide range of Synopsys online services including software downloads, documentation on the Web, and “Enter a Call to the Support Center.”

To access SolvNet, go to the SolvNet Web page at the following address:

https://solvnet.synopsys.com

If prompted, enter your user name and password. If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.

If you need help using SolvNet, click HELP in the top-right menu bar or in the footer.

Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a call to your local support center from the Web by going to http://solvnet.synopsys.com (Synopsys user name and password required), and then clicking “Enter a Call to the Support Center.”
- Send an e-mail message to your local support center.
  - E-mail support_center@synopsys.com from within North America.
  - Find other local support center e-mail addresses at http://www.synopsys.com/Support/GlobalSupportCenters/Pages
- Telephone your local support center.
  - Call (800) 245-8005 from within the continental United States.
  - Call (650) 584-4200 from Canada.
  - Find other local support center telephone numbers at http://www.synopsys.com/Support/GlobalSupportCenters/Pages

1

Introduction to Design Vision

The Design Vision tool is the graphical user interface (GUI) for the Synopsys logic synthesis environment. Design Vision provides analysis tools for viewing and analyzing your design at the generic technology (GTECH) level and the gate level. It also provides all of the synthesis capabilities of the Design Compiler tool. Design Vision provides menu commands and dialog boxes for the most commonly used synthesis features. In addition, you can enter any dc_shell command on the command line in the GUI or the shell.

This chapter contains the following sections:

- Features and Benefits
- User Interfaces
- Methodology
- Supported Formats
- Licensing
- Supported Platforms
- Design Vision and Other Synopsys Products

Features and Benefits

The Design Vision tool provides the following features:

- Window- and menu-driven interface to Synopsys logic synthesis (Design Compiler)
- Analysis visualization capabilities that include
    - A hierarchy browser for navigation through the design hierarchy and exploration of design structures, including viewing area attributes
    - Histograms for visualizing trends in various metrics, for example slack and capacitance
    - Timing analysis driver and path inspector for detailed timing path analysis
    - Path schematics for visually examining timing paths, fanin and fanout logic, and net connectivity
    - Path profiles for visually examining the contributions of different cells and nets to the total delay of a timing path
    - Design schematics and symbol views for visually examining high-level and low-level design connectivity
    - A properties viewer for viewing object properties and editing attribute values
    - Layout views for viewing and analyzing floorplan elements in a design that you optimize by using Design Compiler topographical technology
    - A congestion map for visually examining highly congestion areas in the physical layout
- Reporting capabilities that correlates reported objects to graphical views
- Command-line interface integrated in the GUI with scripting support for all Design Compiler tool command language (dctcl) commands

Using the features of Design Vision, you can

- Invoke Design Compiler synthesis and reporting
- Explore design structures
- Visualize overall timing with Design Vision graphical analysis
- Perform timing analysis for blocks you are synthesizing
- Perform detailed visual analysis of timing paths and connected logic
- Visually examine and debug certain physical floorplan constraints and global routing congestion in Design Compiler Graphical

User Interfaces

Design Vision offers menus and dialog boxes for important Design Compiler functions. Menus also offer visual analysis features and other design viewing features that are not Design Compiler functions. Design Vision also includes a command-line interface that provides access to all the capabilities of Synopsys synthesis tools.

The Design Vision GUI and command-line interface use the dctcl command language. For details, see Design Vision online Help.

Methodology

Design Vision allows you to use the same design methodology and scripts you currently use and to extend your methodology with Design Vision visual analysis. Many Design Compiler commands are available on Design Vision menus. All Design Compiler functions are available through the Design Vision command-line interface.

Supported Formats

Design Vision stores design data in an internal database format. It supports two design database formats: .ddc and Milkyway. The database formats are described below:

- .ddc format

  The .ddc format is a single-file, binary format. The .ddc format stores design data in an efficient manner than the .db format, enabling increased capacity. In addition, reading and writing files in .ddc format is faster than reading and writing files in .db format. The .ddc format stores only logical design information.

- Milkyway format

  The Milkyway format allows you to write a Milkyway database within Design Vision for use with other Synopsys Galaxy tools, such as IC Compiler. The Milkyway format stores both logical and physical design information, but it requires a mapped design.

The Milkyway format is available only when you start Design Vision in topographical mode. You can use the `write_milkyway` command to save netlist and physical design data in a Milkyway design library. You can use a single Milkyway library across the entire Galaxy flow. For more information, see the *Design Compiler User Guide*.

Note:
Design Vision does not support the `read_milkyway` command.

Design Vision can access all the files supported by Design Compiler. Table 1-1 shows the supported design file formats. All netlist formats except .db, equation, PLA, state table, Verilog, and VHDL require special license keys.

Table 1-1 Supported File Formats

Data	Formats
Netlist	Milkyway
	Programmable logic array (PLA)
	Synopsys equation
	Synopsys state table
	Synopsys dc_shell database format (.ddc)
	Verilog
	VHDL
Timing	Standard Delay Format (SDF)
Command Script	dctcl
Library	Synopsys internal library format (.lib)
	Synopsys database format (.db)
Parasitics	dc_shell command scripts

Licensing

To use Design Vision, you need the Design-Vision license. To use Design Vision in topographical mode, you need a Design-Vision license, a DesignWare license, and the DC Ultra package. To use the Design Compiler Graphical layout window in topographical mode, you also need a DC-Extension license. If you use the Milkyway flow in topographical mode, you also need a Milkyway-Interface license; this license is included in the DC Ultra package.

Synopsys licensing software and the documentation describing it are now separate from the tools that use it. You install, configure, and use a single copy of Synopsys Common Licensing (SCL) for all Synopsys tools. Because SCL provides a single, common licensing base for all Synopsys tools, it reduces licensing administration complexity, minimizing the effort you expend in installing, maintaining, and managing licensing software for Synopsys tools.

For complete Synopsys licensing information, see the Common Licensing Installation and Administration Guide. This guide provides detailed information on SCL installation and configuration, including examples of license key files and troubleshooting guidelines.

Supported Platforms

Design Vision is supported on the same platforms that support Design Compiler and the other synthesis tools. For details, see the Installation Guide.

Your hardware and operating system vendor has required patches available for your system. For more information about the supported platforms and the operating system patches necessary to run Synopsys software on supported platforms, go to

http://www.synopsys.com/products/sw_platform.html

From this Web page you can navigate to the Supported Platforms Guide for your release.

Design Vision and Other Synopsys Products

As a visual analysis tool and GUI for Synopsys synthesis, Design Vision works with Design Compiler to synthesize and analyze your design.

The Design Vision and Synopsys PrimeTime tools have similar timing visualization features; however, the tools have different timing engines and differ in their application to analysis. Design Vision has the same static timing engine as Design Compiler. Use Design Vision to perform timing analysis and modification of blocks you are synthesizing. Use PrimeTime for static timing sign-off or for analyzing the timing of a chip or of large portions of a chip.

2

Before You Start

Design Vision is the graphical user interface (GUI) for Design Compiler and the Synopsys synthesis environment. You can use it for visualizing design data and analyzing results. The Design Vision GUI provides a variety of visualization and analysis features, including windows, views, menus and toolbars, interactive mouse tools, and a status bar. Before you start using Design Vision to analyze and troubleshoot a design, you should become familiar with the operation of the GUI and the various tools that it provides.

The following sections provide the general and specific information you need to know before you use Design Vision for the first time:

- The Design Vision Documentation Set
- Running Design Vision
- Exploring the Graphical User Interface
- Using Timing and Design Analysis Views
- Examining Design Information
- Using DFT Analysis Tools
- Viewing Man Pages
- Changing the Appearance of Schematics
- Setting GUI Preferences

The Design Vision Documentation Set

You can find most of what you need to know to run Design Vision in the Design Vision documentation set.

The Design Vision documentation set is divided into these parts:

- *Design Compiler Tutorial Using Design Vision*
- *Design Vision User Guide*
- Design Vision online Help

Other sources of information include man pages, the SolvNet knowledge base, and the Customer Support Center. For information about accessing these sources of information, see “Customer Support” on page xi.

Design Compiler Tutorial Using Design Vision

The *Design Compiler Tutorial Using Design Vision* is an introduction to ASIC design using Synopsys Design Compiler and Design Vision.

The tutorial is a group of guided exercises in which you use Design Compiler to synthesize and analyze a simple design. The tutorial introduces Design Vision as the GUI for synthesis and provides some post-synthesis analysis that takes advantage of the Design Vision visual representation of analysis results.

Use the *Design Compiler Tutorial Using Design Vision* if you are new to Design Compiler. However, even if you are an experienced Design Compiler user, the tutorial is a useful introduction to the features of Design Vision.

In the tutorial, you can read about and perform exercises for basic synthesis and analysis. For example, you

- Read a design from disk into the tool’s workspace
- Define design constraints to direct synthesis results
- Compile the design to gates
- View analysis results
- Modify the design
- Resynthesize the design
- Reanalyze the design

Design Vision User Guide

The *Design Vision User Guide* assumes you are familiar with basic Design Compiler concepts.

The user guide provides guidance in solving particular problems. For example, it presents short procedures that use the analysis visualization features of Design Vision to locate and solve timing problems.

Sometimes steps in a procedure refer to actions without further explanation: for example, “Create a histogram” or “Create a path schematic.” Such steps refer to features of Design Vision that are explained in the Design Vision online Help.

The *Design Vision User Guide* does not contain specific information about individual menu items or dialog boxes. For such information, see the Design Vision online Help.

In Chapter 3, “Performing Basic Tasks,” experienced Design Compiler users can learn how to do certain familiar synthesis tasks using Design Vision. However, the user guide explains such topics only briefly.

Design Vision Online Help

The Design Vision online Help system is available in the Design Vision GUI. You can access the Help system from the Help menu in the Design Vision window. The Help system contains topics that explain the details of tasks that you can perform. For example, if you need help performing a step in a procedure presented in the user guide, you can find the information you need in the online Help system.

Information in online Help is grouped in the following categories:

- Feature topics

  Overviews of Design Vision window components and tools.

- How-to topics

  Procedures for accomplishing synthesis and analysis tasks.

- Reference topics

  Explanations of views, toolbar buttons, menu commands, and dialog box options.

Note:
> Before you can access online Help from within Design Vision, the Web browser executable file must be listed in your UNIX or Linux path variable.

Design Vision online Help is a browser-based HTML Help system designed for viewing in the Firefox and Mozilla Web browsers.

To access online Help in Design Vision,

1. Choose Help > Online Help.

   The Web browser appears and displays the Welcome topic for the Design Vision Help system.

2. Use the navigation frame (leftmost frame) to find the information you need in one of the following ways:

   - Find the topic in the hierarchical organization of the Help system by clicking Contents and expanding the appropriate books until you find the information you need.
   - Find the topic by its subject by clicking Index and looking for the subject in the alphabetical listing.
   - Search for key words found in the topic by clicking Search and entering the keywords.

     If more than one topic has the words you are searching for, you must select the appropriate topic from a list of topics.

You can view Design Vision online Help as a stand-alone Help system in your Web browser by opening the file named index.html in the online Help directory: $SYNOPSYS/doc/syn/html/dvoh/enhanced.

Design Vision online Help makes extensive use of JavaScript and cascading style sheets (CSS). If your browser encounters problems displaying online Help, open the browser preferences and make sure that JavaScript and style sheets are enabled and that JavaScript is not blocked by your security preferences.

Note:

> If you reset preferences while the Help system is open, you might need to click the Reload button on the browser's navigation toolbar after you reset the preferences.

You can use the following browsers to view the Design Vision online Help system:

- Firefox version 1.5 or 2.0
- Mozilla version 1.7

The default online Help browser is Firefox. You can use the `gui_online_browser` variable to control which browser Design Vision uses to display online Help.

To set the browser for the current GUI session,

- Enter the following command after you start the Design Vision GUI:

  ```
  set gui_online_browser "browser-name"
  ```

  The *browser-name*: can be `firefox` or `mozilla`.

Alternatively, you can use a setup file to set a default online Help browser for any GUI session. Create a file named .synopsys_dv_gui.tcl in your home directory, and enter the `set` command in the file:

```
set gui_online_browser "browser-name"
```

Running Design Vision

Design Vision offers two interfaces for synthesis and analysis: the Design Vision graphical user interface (GUI) and a shell command-line interface.

- The Design Vision GUI is an advanced visualization and analysis toolset.

  Design Vision can perform certain tasks, such as very accurately displaying your design, and it provides visual analysis tools that are available only in the GUI. The look and feel of the Design Vision GUI is consistent with the look and feel of other Synopsys GUIs.

- The Design Vision shell is a text-only environment that is identical to the Design Compiler shell command-line interface (dc_shell).

  You enter commands at the command-line prompt the same way you enter them in dc_shell. For information on using the shell command-line interface, see the *Design Compiler User Guide*.

The shell command-line interface is always available. You can open or close the GUI multiple times during a session. The GUI opens by default when you start Design Vision. Help is available for both interfaces. For details, see "Viewing Man Pages" on page 2-28 and "Design Vision Online Help" on page 2-3.

The GUI provides menu commands and toolbar buttons that you can use to visualize design data and analyze results. You can also use GUI menu commands and dialog boxes to run many basic or frequently used Design Compiler commands. In addition, the GUI provides a command console with a command-line interface and views of the session log and the command history. You can perform any task in the GUI that you can perform in the shell.

Design Vision uses the Design Compiler tool command language (dctcl). The command language provides capabilities similar to UNIX command shells, including variables, conditional execution of commands, and control flow commands. You can extend the Design Compiler command language by writing reusable procedures and scripts. For details, see the Using Tcl With Synopsys Tools manual.

You can execute dctcl commands in the following ways:

- By typing single commands interactively on the console command line in the Design Vision window
- By entering single commands interactively in the shell

- By running one or more command scripts, which are text files of commands

Using this approach allows you to supplement the subset of Design Compiler commands available through the menu interface.

To learn more about running Design Vision, see the following sections:

- Setup Files for Design Vision
- Starting a Design Vision Session
- Opening and Closing the GUI
- Exiting Design Vision
- Using dctcl Scripts
- Getting Command-Line Help
- Using Command Log Files
- Using the Filename Log File

Setup Files for Design Vision

When you start Design Vision, it automatically executes commands in the three standard Design Compiler setup files that dc_shell uses. These files have the same names, .synopsys_dc.setup, but reside in different directories. The same sourcing rules apply for both Design Vision and dc_shell. For more information about the .synopsys_dc.setup files and the initialization settings for synthesis, see the Design Compiler User Guide.

In addition, Design Vision reads another set of setup files when you open the GUI, named .synopsys_dv_gui.tcl. You can use these files to perform GUI-specific setup tasks. Use the .synopsys_dc.setup files to perform non-GUI application setup tasks. Settings from the .synopsys_dv_gui.tcl files override settings from the .synopsys_dc.setup files.

Design Vision reads the .synopsys_dc.setup and .synopsys_dv_gui.tcl files from three directories in the following order:

- The Synopsys root directory

  These files contain system variables defined by Synopsys and general Design Compiler setup information for all users at your site. Only the system administrator can modify these files.

- Your home directory

  Use these files to set variables that define your synthesis working environment. The variables in these files override the corresponding variables in the systemwide setup files.

- The current working directory (the directory from which you start Design Vision)

  Use these files for project- or design-specific variables that affect all designs in the project directory. To use these files, you must start Design Vision from this directory. Variables defined in these files override the corresponding variables in the user-defined and systemwide setup files.

In addition, you can use the setup file in your home or design directory to define dctcl scripts that you need to run during a Design Vision session. For more information, see “Using dctcl Scripts” on page 2-10.

Besides reading the setup files, Design Vision also loads preferences and view settings from a file named .synopsys_dv_prefs.tcl in your home directory. You should not edit this file. For more information, see “Setting GUI Preferences” on page 2-30.

For more information about the locations of setup files and initialization settings for synthesis, see the Design Compiler User Guide.

Starting a Design Vision Session

Design Vision operates in the X Windows environment on UNIX or Linux. Before starting Design Vision, make sure your $SYNOPSYS variable is set and the path to the bin directory is included in your $PATH variable. Before starting the GUI, make sure your $DISPLAY environment variable is set to the name of your UNIX or Linux system display.

Design Vision provides an option to enable the Design Compiler topographical technology for physically aware RTL synthesis. You can start a Design Vision session in topographical mode, but you cannot enable or disable the mode during a session. For details about using topographical mode, see the Design Vision online Help.

Note:

> You can query the mode by running the `shell_is_in_topographical_mode` command. The command returns 1 if Design Vision is running in topographical mode; otherwise it returns 0.

To start Design Vision, enter the `design_vision` command in a UNIX or Linux shell:

```
% design_vision
```

If you are using Design Compiler topographical technology or Design Compiler Graphical, you must indicate this by specifying the -topographical_mode option. You can abbreviate this option to as short as -to. For example, enter

```
% design_vision -topographical_mode
```

Topographical mode requires a DC Ultra license and a DesignWare license. For information about additional license requirements, see “Licensing” on page 1-4. For information about using Design Vision in topographical mode, see “Using Design Compiler Topographical Technology in Design Vision” on page 3-10.

These commands start the Design Vision in the GUI by default. The Design Vision window appears on the screen, and the command-line prompt, which is design_vision> by default or design_vision-topo> in topographical mode, appears in the shell and in the command console. By default, the console is located above the status bar near the bottom of the Design Vision window.

You can also start Design Compiler in the GUI. Enter the dc_shell command with the -gui option. For example, you can enter either of the following commands:

```
% dc_shell -gui
% dc_shell -topographical_mode -gui
```

Design Compiler starts in dc_shell by default. For more information about starting Design Compiler, see the *Design Compiler User Guide*.

You can start the Design Vision in the shell command-line interface without opening the GUI by specifying the -no_gui option. For example, enter one of the following:

```
% design_vision -no_gui
% design_vision -topographical_mode -no_gui
```

The appropriate Design Vision command-line prompt appears on the shell command line. When you want to open the GUI, enter the gui_start command. For more details about opening and closing the GUI, see “Opening and Closing the GUI” on page 2-9.

The design_vision command provides startup options for customizing your session. For example, you can run a batch startup script, initiate certain dctcl commands, or check out additional licenses.

You can use

- -f to execute the commands in the specified dctcl script file before displaying the initial design_vision shell prompt or the Design Vision window
- -x to execute a dc_shell command at startup
- -checkout to access licensed features in addition to the features checked out by default

To see the complete list of available options without starting Design Vision, enter

```
% design_vision -h
```

For detailed information about the startup options, see the Design Compiler Command-Line Interface Guide or the man page for the `design_vision` command.

At startup, Design Vision performs the following tasks:

1. Creates a command log file.
2. Reads and executes commands from the dc_shell setup files and GUI preferences from the preferences file in your home directory.
3. Executes any commands specified by the `-x` and `-f` options.
4. Displays the program header and shell prompt in the UNIX or Linux shell where you invoked the tool. The program header lists all features for which your site is licensed.

When you open the GUI, either at startup or from within the design_vision shell, Design Vision performs the following tasks:

1. Reads and executes commands from the Design Vision GUI setup files.
2. Opens the Design Vision window.

Opening and Closing the GUI

If you start a Design Vision session without the GUI, you can open the GUI from the shell command line. You can open or close the GUI without exiting Design Vision at any time during the session.

To open or reopen the GUI from the Design Vision shell, enter the following command:

```
design_vision> gui_start
```

To close the GUI without exiting Design Vision, do either of the following:

- Choose File > Close GUI.
- Enter the following command:

  ```
  design_vision> gui_stop
  ```

Exiting Design Vision

To exit Design Vision, you can do any of the following:

- Choose File > Exit, and then click OK in the message box that appears.
- Enter `exit` or `quit` on the command line.
- Press Control-c three times in the UNIX or Linux shell.

Design Vision does not automatically save the designs loaded in memory. For information about how to save your design, see "Saving the Design" on page 3-7.

Using dctcl Scripts

Designers often use scripts to accomplish routine repetitive tasks such as setting constraints or defining other design attributes. You can use your existing dctcl scripts in Design Vision. You can define scripts in your setup files and in separate script files.

To run scripts in Design Vision,

- Choose File > Execute Scripts.

  The Execute File dialog box opens. Use the dialog box to navigate to the appropriate directory and run your script.

Alternatively, you can run scripts from the design_vision command line by using the `source` command.

Table 2-1 is a short list of some commands that are useful when you are writing scripts for use with Design Vision. For more information about a command, see its man page.

Table 2-1 Commands for Scripting

Command	Type	Description
`get_selection`	dctcl	Returns a collection that is currently selected in the GUI
`change_selection`	dctcl	Changes the selection in the GUI to the collection passed to it as a parameter
`gui_create_schematic`	dctcl	Generates a design schematic or a symbol view of the selected design or cell

Getting Command-Line Help

Design Vision provides three levels of command help:

- A list of commands
- Command usage help
- Topic help

To get a list of all dctcl commands, enter the command:

```
design_vision> help
```

To get help about a particular dctcl command, enter the command name with the `-help` option. The syntax is

```
design_vision> command_name -help
```

To get topic help for a dctcl command, variable, or variable group, enter

```
design_vision> man topic
```

Replace *topic* with the name of a dctcl command, variable, or variable group. By using the `man` command, you can display the man pages for the topic while you are interactively running Design Vision. In the GUI, you can view topic help in the man page viewer by choosing Help > Man Pages. For details, see "Viewing Man Pages" on page 2-28.

Using Command Log Files

The command log file records the dctcl commands processed by Design Vision, including setup file commands and variable assignments. By default, Design Vision saves the command log to a file named command.log in the directory where you started the tool.

You can change the name of the command log file by using the `sh_command_log_file` variable in your.synopsys_dc.setup file. If neither your user-defined nor project-specific .synopsys_dc.setup file contains this variable, Design Vision automatically creates the command log file with the name command.log.

Each Design Vision session overwrites the command log file. To save a command log file, move it or rename it. You can use the command log file to

- Produce a script for a particular synthesis strategy
- Record the design exploration process
- Document any problems you are having

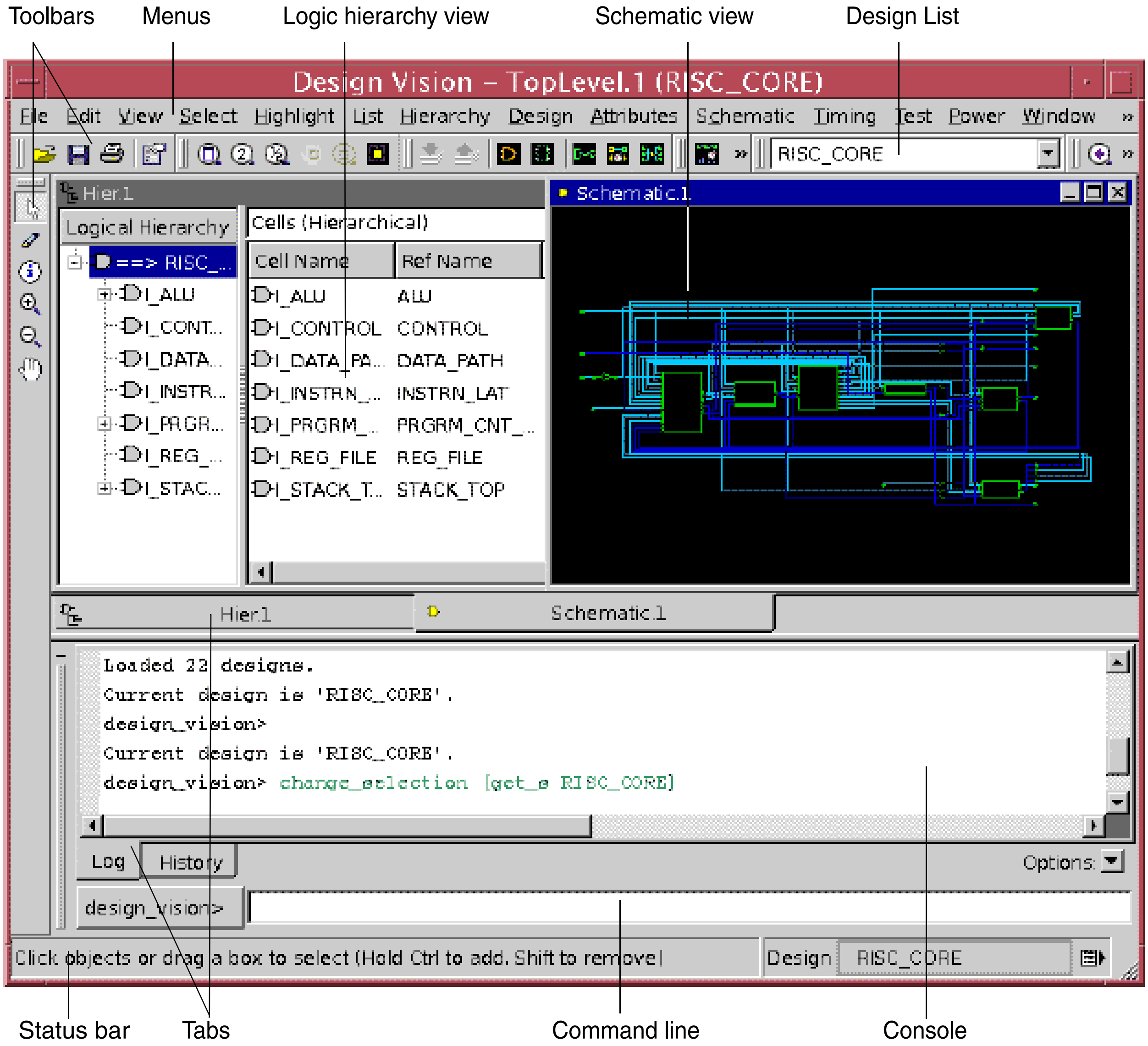
Using the Filename Log File

By default, Design Vision creates a file name log of the files it has read, and saves the log in a file named filename log in the directory where you started the tool. You can use this file to identify data files needed to reproduce an error in case Design Vision terminates abnormally. You can specify a different name for this file by using the `filename_log_file` variable in your.synopsys_dc.setup file.

Exploring the Graphical User Interface

The Design Vision window appears by default when you start Design Vision. Figure 2-1 shows the Design Vision window you see when you start Design Vision, read in a design, and open a design schematic for the top-level design.

Figure 2-1 The Design Vision Window



Cell Name	Ref Name
I_ALU	ALU
I_CONTROL	CONTROL
I_DATA_PA...	DATA_PATH
I_INSTRN_...	INSTRN_LAT
I_PRGRM_...	PRGRM_CNT_...
I_REG_FILE	REG_FILE
I_STACK_T...	STACK_TOP

```
Loaded 22 designs.
Current design is 'RISC_CORE'.
design_vision>
Current design is 'RISC_CORE'.
design_vision> change_selection [get_s RISC_CORE]
```

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)

The Design Vision window consists of a title bar, a menu bar, and several toolbars at the top of the window and a status bar at the bottom of the window. The title bar and menu bar are always visible. You can display or hide individual toolbars or the status bar.

You use the workspace area between the toolbars and the status bar to display view windows and panels. View windows provide graphic or textual views of design information. Panels provide interactive tools for setting options or performing often used tasks. View windows and panels can contain tabs with multiple views or pages. The “active view” is the view that has the mouse focus.

The hierarchy browser (logic hierarchy view) and the console appear in the Design Vision window by default. For more information about these features, see the following sections. For additional details, see the Design Vision online Help.

- Design Vision Windows
- The Hierarchy Browser
- The Command Console

Design Vision Windows

The Design Vision GUI displays information in top-level GUI windows that you can move, resize, minimize, or maximize by using the window management tools on your UNIX or Linux desktop. The title bar lists the product name (Design Vision) of the tool you are using, the name of the window you are viewing, and the name of the active view. The window name includes the unique number of the window.

Design Vision provides the following top-level GUI windows:

- The Design Vision window appears automatically when you start a Design Vision session or open the GUI.
- The layout window is available for visualizing the physical aspects of a design in Design Compiler Graphical when you start Design Vision in topographical mode.

You can open multiple instances of the Design Vision window or the layout window and use them to compare views, or different design information within a view, side by side.

You can resize, move, and minimize or maximize each window. The windows are numbered sequentially throughout the session. All open instances of a top-level GUI window share the same designs in memory and the same current timing information. However, you can configure the toolbars, status bar, view windows, and panels independently for each window.

For more information about the Design Vision window, see the following sections:

- Menus
- Toolbars
- View Windows and Panels
- Status Bar

Menus

The menu bar contains menus with the commands you need to use the GUI. To use a menu bar menu, click the menu name and choose the command. You can display a brief message in the status bar about the action that a menu command performs by holding the pointer over the command. For menu commands that can also be used by pressing a toolbar button or typing a keyboard shortcut, the menus show representations of those alternatives.

Note:

> If the window is not wide enough to display all the menu names on the menu bar, the window displays all the menu names that fit, from left to right, followed by an overflow button (»). To access the other menus, click the overflow button.

Some frequently used menu commands are also available on pop-up menus for individual views. To use a pop-up menu, right-click and choose a command.

Toolbars

Toolbars provide quick access to frequently used commands and interactive operations. To determine the function of a toolbar button, hold the pointer over the button. A ToolTip displays the name of the button, and the status bar displays a brief description of its use. You cannot disable these messages.

Toolbars are always attached to a window edge. By default, the toolbars appear in a row at the top of the window below the menu bar. You can enhance your working environment by moving individual toolbars to different positions below the menu bar, or to the left, right, or bottom edge of the window. You can also disable a toolbar, hiding it from view.

If a window edge is not long enough to display all of the toolbars attached to it, the GUI displays the full toolbars that fit and shortened versions of the other toolbars. A shortened toolbar consists of a default toolbar button and an overflow button (»). To access the other toolbar buttons on a shortened toolbar, click the overflow button.

View Windows and Panels

View windows are child windows that display graphic or textual views of design information inside a top-level GUI window. View windows that contain multiple views provide a tab for each view. When you open a view window that has multiple views, it displays a default view. To change to a different view, you click its tab. When you click anywhere within a view window, Design Vision highlights its title bar to indicate that it has the focus (that is, it is the active view) and can receive keyboard and mouse input.

The Design Vision window provides the following types of view windows:

- Graphic views (graphical descriptions of design information such as histograms and schematics)

- Hierarchy views (for traversing hierarchical structures and gathering design information at different hierarchy levels)
- Text views (textual design information such as reports and object lists)

When you start Design Vision, a logic hierarchy view appears in the workspace. The analysis tasks that you perform during the session determine which other view windows you open. For more information about the logic hierarchy view, see "The Hierarchy Browser" on page 2-17.

Panels are enhanced toolbars that contain tools for setting options or performing frequent tasks while working with the design in view windows. Most panels are associated with a particular view window and operate on the active view (the view that has the mouse focus). (An exception is the console, which contains a command line and its own views.) A tabbed panel contains tabs that you can click to access different tools or views. The first time you open a panel during a session, it displays the tools or view for its default tab.

The Design Vision window provides the following panels:

- Console (provides dctcl command line and command log and history views)
- View Settings (provides options for setting display properties in schematic views)
- Query Objects (displays object information when you use the Query tool)

By default, when you start Design Vision, the console is docked to the bottom edge of the Design Vision window and the other panels are hidden. For more information about the console, see "The Command Console" on page 2-18.

You can adjust the sizes of view windows and panels for viewing purposes, and you can move them to different locations within the workspace area. In addition, you can

- Arrange the open view windows by tiling or cascading them within the workspace area
- Minimize individual view windows, or maximize a view window to fill the workspace area
- Dock or undock individual panels by attaching them at edges of the Design Vision window or separating them from the edge so they can float inside or outside the workspace.

The GUI displays a tab at the bottom of the workspace area for each open view window. When you click a tab, the GUI displays its view window on top of the other view windows and makes it the active view. If a view window and a panel overlap on the screen, the panel appears on top of the view window.

Status Bar

The status bar, located at the bottom of the Design Vision window, displays the information listed in Table 2-2.

Table 2-2 Information Displayed by the Status Bar

When you do this	The status bar displays this information
Select one object	Object name
Select multiple objects	Number of selected objects
Hold the pointer over a menu command, toolbar button, or tab	Information about the action it performs

You can quickly display the list of selected objects in the Selection List dialog box by clicking the ▤▶ button at the right end of the status bar.

You can hide or display the status bar by choosing View > Status Bar.

The Hierarchy Browser

The logic hierarchy view appears when you start Design Vision and open the GUI. Use it to navigate through your design and gather information. The view is divided into the following two panes:

- Instance tree
- Objects list

The instance tree lets you explore the design hierarchy and see the relationships among its levels. If you select the instance name of a hierarchical cell (an instance that contains subblocks), information about that instance appears in the object table. You can Shift-click or Control-click instance names to select combinations of instances.

By default, the object table displays information about hierarchical cells belonging to the selected instance in the instance tree. To display information about other types of objects, select the object types in the list above the table. You can display information about hierarchical cells, all cells, pins and ports, pins of child cells, and nets.

If some of the text in a column is missing because the column is too narrow, you can hold the pointer over it to display the information in an InfoTip. You can also adjust the width of a column by dragging its right edge left or right.

For more details about the logic hierarchy view, see the “Browsing the Design Hierarchy” topic in online Help.

The Command Console

When you start Design Vision and open the GUI, the console panel is docked above the status bar by default. You can use the console to

- Enter Tcl commands on the console command line
- View either the session transcript (log view) or the command history list (history view) by clicking the tabs above the command line
- Copy and edit or reuse commands, and search for, select, and save commands or messages

You can enter any dc_shell command on the console command line. Enter these commands just as you would enter them at the dc_shell prompt in a standard UNIX or Linux shell. For example, if you enter `get_selection`, the console log view displays a list of the names of all selected objects.

You can display or hide the console, and you can increase or decrease its height. You can dock it to the top or bottom edge of the Design Vision window, or you can undock it and resize it or move it around or off its Design Vision window. For more information about the Console, see the following sections:

- Console Log and History Views
- Console Command-Line Editing

Console Log and History Views

The console displays information about the commands you use during the session in the following views:

- Log view
- History view

To select a view, click the tab above the command line. The log view is displayed by default when you start Design Vision.

The log view displays a transcript of session information, which includes the commands you entered and the Design Vision output and messages resulting from your commands. You can use this information to

- Check on tool status after performing functions

- Troubleshoot problems that you encounter
- Search the transcript for information on past functions
- Save the transcript, selected text, or just the error and warning messages in a text file

You can reenter commands you have already used by copying them from the log to the command line.

The history view lists menu, dialog box, and dc_shell commands you have used in the current session. You can do the following with this list:

- See which commands you have used
- Find and reuse commands already used
- Save the list for later use

For more details about the console log and history views, see the “Viewing the Session Log” and “Viewing the Command History” topics in online Help.

Console Command-Line Editing

When you issue a command (by pressing Return or clicking the prompt button to the left of the command line), Design Vision echoes the command output (including processing messages and any warnings or error messages) in the console log view

You can display, edit, and reissue commands on the console command line by using the arrow keys to scroll up or down the command stack and to move the insertion point to the left or right on the command line. You can copy text in the log view and paste it on the command line the same way you would in a UNIX or Linux shell, by selecting the text with the left mouse button and pasting it with the middle button. You can also select commands in the history view and edit or reissue them on the command line.

If you need to enter a command or dctcl procedure that uses multiple lines, you can vertically expand the command line by either typing a backslash (\) at the end of a line and pressing Return or clicking in the command line and pressing Shift-Return. The command line automatically shrinks to a single line when you issue the command.

For more details about console command-line editing, see the “Entering Commands in the Console” topic in online Help.

Using Timing and Design Analysis Views

Design Vision provides views you can use for both global timing analysis and detailed analysis of individual timing paths and object connectivity. For information about the different kinds of timing and design analysis views you can use, see the following sections:

- Histogram Views
- Timing Analysis Driver
- Schematic Views
- Path Profile Views
- Path Inspector Windows

Histogram Views

Histograms provide high-level views of design timing for visual timing analysis. You can use histograms to view the overall timing performance of your logic design and to select individual timing paths for further study in timing reports or other analysis views. Design Vision provides the following predefined histograms: endpoint slack, path slack, and net capacitance.

- Endpoint slack histograms show a distribution of timing slack values for all endpoints in the design.

  You can choose maximum or minimum delay (setup or hold). The slack distribution provides an overall picture of how close the design is to meeting requirements.

- Path slack histograms show a distribution of timing slack values for selected paths or for the paths with the worst slack in the design.

  You can select a maximum or minimum delay type (setup or hold), set the maximum number of paths and the number of worst paths per endpoint, and select a path group. You can also specify individual paths to, from, or through selected objects (similar to the way you specify paths for timing reports).

- Net capacitance histograms show a distribution of net capacitance values for selected nets or for all nets in the design.

When you use the timing analysis driver window to view timing path details, you can also generate histograms that show the distribution of values for certain types of path details listed in the window.

For more details about histogram views, see the “Viewing Histograms” topic in online Help.

Timing Analysis Driver

The timing analysis driver is the focal point for timing analysis in Design Vision. You can examine timing path details, such as attribute values, generate timing reports and histograms, and display selected paths in path schematics and timing inspector windows.

The timing analysis driver window contains a timing path table, a button bar, and a command display box.

- The timing path table displays a list of paths that you specify. The table columns show the startpoint name, endpoint name, and other details about each path.
- The button bar below the table provides buttons that you can use to
  - Generate a histogram for the value distribution of a specific attribute
  - Select one or more paths and generate a timing report
  - Select and display one or more paths in a path schematic
  - Select and display an individual path in a path inspector window.

  You can also load a different collection of timing paths, save the path details in a text file, and hide or display individual table columns.
- The command display box shows the command and options used to find the paths. You can copy text in the command display box and paste it somewhere else, such as on the console command line.

When you open the timing analysis driver view window, or reload the paths if the window is already open, you use the Select Paths dialog box to load a collection of paths into the timing path table. You can

- Set options in the dialog box and run the `get_timing_paths` command
- Select and run a predefined collection command

  You can load all selected paths or all highlighted paths.
- Enter a command to define and load a custom collection

  The tool adds the command to the list of predefined commands.

  If you define a custom collection in a variable, you can use the `get` command to load the paths from the collection.

The dialog box options are set by default to select the 20 timing paths with the worst slack times in the design. You can reset the dialog box options to their default values by clicking the Default button.

For more information about the timing analysis driver, see the ”Analyzing Timing Path Details” topic in online Help.

Schematic Views

Schematic views show graphical representations of the logic design. Symbol views provide graphic overviews of logic designs. Design Vision provides the following types of schematic views:

- Path schematics that display individual timing paths or selected design logic (cells, ports, or nets) in a flat, single-sheet schematic

  A path schematic can contain cells, pins, ports, and nets. Hierarchy crossings indicate places where the path traverses the design hierarchy. You can display an InfoTip showing information about an object by holding the pointer over the object. The type of information displayed depends on the object type.

  You can generate a path schematic to display timing paths, selected logic, or the fanin or fanout logic for selected objects. You can display or hide logic in a path schematic. You can also add timing paths, add fanin or fanout logic levels, or remove selected logic.

- Design schematics that display a design as a schematic composed of cell instances, pins, nets, and ports

  A design schematic shows both leaf-level logic (gates) and subdesigns (blocks). You can move up or down in the design hierarchy to view the schematic for each block in the hierarchy.

You can use schematic and symbol views to visually analyze timing and logic in the optimized design and to gather information that can help you guide later optimization operations. Objects you select in a schematic or symbol view are cross-selected in other views. This capability allows you to efficiently analyze the logic and timing aspects of your design.

The View Settings panel provides options you can use to change display characteristics in the active schematic or symbol view, including object colors, text colors, text sizes, and text visibility. You can save the new settings in your preferences file, or reload settings previously saved in the file. For details, see See “Changing the Appearance of Schematics” on page 2-29.

For more information about schematic views, see the “Using Path Schematics” and “Using Design Schematics” topics in online Help.

Path Profile Views

Path profile views help you examine the contributions of individual cells and nets to the total delay of a timing path.

The view window contains a table that shows path and pin data. For each path, the table displays the path name, total delay time, relative pin delay contributions, and full path name. For each pin on each path, the table displays the pin name, individual delay time, relative contribution, clock edge (rising or falling), and full pin name. The combined delays for each path and the relative delay contribution for each pin appear graphically in bar graphs that represent the percentages of the total path delay.

For more information about path profile views, see the ”Viewing Path Profiles” topic in online Help.

Path Inspector Windows

Path inspector windows let you examine detailed information about selected timing paths. When you select a path and load it into a path inspector window, you can view the path status, path and clock summaries, clock network and path data elements, path delay profiles, and a path schematic with clock path highlighting. The window provides a variety of tools for viewing different aspects of the selected path.

A path inspector window has its own menu bar, toolbars, and status bar. You can move or resize the window and manipulate the toolbars and status bar the same way you do with the Design Vision window. The path information appears on three panels between the toolbars and the status bar:

- The Status panel displays the path startpoint, endpoint, delay type, and slack. It also provides controls for automatically or manually loading another path.
- The Tab panel displays path summaries, clock network and path data element tables, and path delay profiles in a set of tabbed pages. You can copy the summaries and paste them into text files, export the tables to text files, customize the element tables, and generate delay calculation reports for selected cells or nets.
- The Schematic View panel displays a path schematic of the complete path (datapath, clock launch path, and clock capture path) with data and clock path overlays. You can display or hide the paths and the overlays. You can also add fanin and fanout logic to the path schematic.

The Status panel contains an automatic follow-selection mechanism that is enabled by default. This mechanism, when enabled, causes the contents of the path inspector window to “follow” the current selection in the Design Vision window. If you select a different path, the path inspector window immediately loads the data for the new path.

For more information about path inspector windows, see the ”Using the Path Inspector” topic in online Help.

Examining Design Information

Design Vision provides tools you can use to view detailed design information. For information about these tools, see the following sections:

- Viewing and Editing Object Properties
- Viewing and Saving Reports
- Viewing Design and Object Lists
- Viewing the Selection List

Viewing and Editing Object Properties

You can use the Properties dialog box to view object properties for selected designs, design objects, or timing paths. You can also set, change, or remove the attribute values for certain properties.

The dialog box lists the object properties in a table with two columns (for property names and property values) and a row for each property. The properties you can view include object names, attribute values, and certain timing values. Timing attribute values do not appear until you perform an operation that updates timing information, such as generating a timing report or opening a histogram.

The list of properties differs from one object type to another. By default, all the properties for an object appear in the table. You can filter the property list for some types of objects by selecting a property group. The predefined property groups can include all, basic, placement, and timing. The placement group is available only in topographical mode. You can also view the properties in a custom attribute group that you define.

When you select multiple objects, the property lists are displayed separately for each object. You can click the previous and next arrow buttons to navigate from one list of object properties to another. Alternatively, you can select an option to list the property values for all the selected objects together in a single table.

For more information about the Properties dialog box, see the “Viewing and Editing Object Properties” and “Using Attribute Groups“ topics in online Help.

Viewing and Saving Reports

Report views display design and timing reports. You can use a report view to view report information and select reported objects. You can also

- Save or append the report in a text file
- Display a report previously saved in a text file

In reports that list object names (cell, net, port, and worst-path timing reports), you can select an object by clicking its name (blue text) in the report view. When you select an object in a report view, Design Vision displays the design schematic for the design in which the object is located and magnifies the schematic to fit the selected object in the view. The name of the selected object also appears in the selection list, and the object is displayed in the selection color (white) in all open schematic views.

For more information about the report view, see the “Viewing Reports” topic in online Help.

Viewing Design and Object Lists

You can generate a list of objects (designs, cells, nets, or ports and pins) and display information about them in a list view. You can display information about

- The designs loaded in memory
- Selected objects (such as selected cells)
- Objects related to other selected objects (such as pins of selected cells)
- Objects with a common function or attribute (such as fixed cells)

You can select some or all of the objects in a list by clicking or dragging the pointer across their names in the list view. (You can also use Shift-click or Control-click to select multiple names.) You can also

- Sort the information alphabetically by clicking a column heading. Click the heading again to reverse the sort.
- Scroll up and down in the table by pressing the Up Arrow and Down Arrow keys.
- Filter the objects listed in the table, limiting it to objects based on a character string or regular expression that you define.

If some of the text in a column is missing because the column is too narrow, you can hold the pointer over it to display the information in an InfoTip. You can also adjust the width of a column by dragging its right edge left or right.

For more information about design and object lists, see the “Viewing Collections of Objects” and “Viewing a List of Designs” topics in online Help.

Viewing the Selection List

The Selection List dialog box displays a list of the names and object types of all selected objects in the current design. When you select objects in other views, their names automatically appear in the selection list.

You can deselect object names and remove their names from the list. You can also filter the selection list, limiting it to information based on a character string or regular expression that you define.

For more information about the selection list, see the “Viewing the Selection List” topic in online Help.

Using DFT Analysis Tools

If you are using DFT Compiler, you can access tools from the Test menu to DRC violations and hold time violations. For information about the test analysis views you can use, see the following sections:

- DRC Violation Browser
- Violation Inspector
- DFT Hold Time Analysis Window

DRC Violation Browser

If you are using DFT Compiler, you can use the DRC violation browser to search for and view information about DFT unified DRC violations in the current design. The violation browser can display both static and dynamic violation messages. Static violations occur as a result of the design topology. For dynamic violations, you can view waveforms for pins on the violated path.

The violation browser window consists of two panes. A violation category tree appears in the left pane. When you select a violation category, a list of violations appears in the right pane.

- The violation category tree groups warning and error messages into categories that help you find the problems you are interested in.
- The violation list displays the violation ID and pin name for each violation in the list.

You can view the man page for a warning or error message by selecting the message and clicking the Help button. Design Vision displays the man page in the man page viewer.

You can view information about a violation pin by selecting the violation. If you want to visually inspect and debug violations by using the violation inspector window, select the violations and click the Inspect Violation button.

The violation inspector is described in the following section. For more information about the DRC violation browser, see the “Examining DRC Violations” topic in online Help.

Violation Inspector

If you are using DFT Compiler, you can analyze and debug DFT unified DRC violations by inspecting them in a violation inspector window. You can inspect one or more violations of the same type. The violation inspector provides both a schematic view (violation schematic) for inspecting static violations and a coordinated waveform view for inspecting dynamic violations.

The violation inspector displays the pin data that corresponds to the most suitable pin data type for debugging the violation. If you need to, you can change to a different pin data type.

- The pin data for static violations is constant; the simulation values do not change over time.
- The pin data for dynamic violations represents simulation values for a series of initialization cycles.

You can analyze and debug static violations by inspecting the design topology. Use the violation schematic to view and probe the signal and clock pins where the violations occur.

The violation schematic is an enhanced path schematic. You can perform any path schematic operation in a violation schematic, including selecting objects, viewing object information, highlighting objects or timing paths, and magnifying and traversing the view.

To debug dynamic violations, you can select pins in the violation schematic and view their simulation values in the waveform view. Simulation values can be constant or they can vary over time in a series of simulation “events.”

The violation inspector displays the pin data that corresponds to the most suitable pin data type for debugging the violation. To simulate the pin data for a dynamic violation, you must change to a pin data type that supports simulation values.

For more information about violation inspector, see the “Inspecting Static DRC Violations” and “Inspecting Dynamic DRC Violations” topics in online Help.

DFT Hold Time Analysis Window

If you are using DFT Compiler, you can use the hold time analysis window to view information about scan cells that have hold time violations. You can also select scan cells in the window for further examination with other analysis tools.

The hold time analysis window contains a scan cell table that displays a list of scan cells that have hold time violations. The table columns show the scan cell names and other details about each cell. A button bar below the table lets you load a different list of scan cells, save the scan cell details in a text file, customize the table columns, and access other analysis tools.

For more information about the DFT Hold Time Analysis Window, see the “Analyzing Hold Time Violations” topic in online Help.

Viewing Man Pages

Design Vision provides an HTML-based browser window that lets you view, search, and print man pages for commands, variables, and error messages. You can use it to

- Display a man page
- Search for text on the man page you are viewing
- Print the man page you are viewing
- Browse back and forth between man pages you have already viewed

To view a man page in the man page viewer,

1. Choose Help > Man Pages.

   The man page viewer appears. The home page displays a list of links for the different man page categories.
2. Click the category link for the type of man page you want to view.

   The choices are Commands, Variables, and Messages. The contents page for the category displays a list of title links for the man pages in that category.
3. Click the title link for the man page you want to view.

Note:
You can also display man pages in the man page viewer by using the `man` command (or the `gui_show_man_page` command) on the console command line.

You can browse back and forth between pages you previously viewed the same way you browse Web pages in a Web browser. For more information about the man page viewer, see the “Viewing Man Pages” topic in online Help.

Changing the Appearance of Schematics

For display purposes only, when Design Vision generates a schematic, it applies the same display characteristics to all the objects or text of a given object type. The default display characteristics work well for most designs. However, if you need to customize schematics for a specific design or environment, you can use the View Settings panel to

- Change the colors for different types of design objects (cells, ports, pins, nets, buses, bus rippers, or hierarchy crossings)
- Change the colors or text sizes for different types of object names or object annotations
- Hide or display for different types of object names or object annotations

The changes apply only to the active view. However, you can save the new settings for use with in new schematic views that you open or in future Design Vision sessions.

To change visual display settings in the active schematic view:

1. Choose View > View Settings to open the View Settings panel if it is not already open.
2. Set the desired options.
3. Click Apply.

To save the new settings in your preferences file,

- On the View Settings panel, choose Options > Save to Preferences.

To load schematic view settings from your preferences file,

- On the View Settings panel, choose Options > Set from Preferences.

You can use the View Settings panel with any schematic: path schematic, design schematic, symbol view, or DRC violation schematic. Each new schematic you open reads in the default display characteristics from your preferences file.

For more information about the View Settings panel, see the “Changing Schematic Display Options” topic in online Help.

Setting GUI Preferences

At the beginning of the GUI session, Design Vision loads GUI preferences from your preferences file. The default system preferences are set for optimal tool operation and work well for most designs. However, if necessary, you can change GUI preferences during the session.

You can set preferences that control how text appears in GUI windows and whether commands for selection or interactive operations appear in the session log. You can also set various global, schematic view, and layout view default controls.

To set GUI preferences,

1. Choose View > Preferences.

   The Application Preferences dialog box appears.

2. Select a category in the Categories tree.

   The page for that category appears.

3. Set options as needed.

4. Click OK or Apply.

When you change preference settings during a GUI session, Design Vision automatically saves the new preference settings in the preferences file named .synopsys_syn_prefs.tcl in your home directory. The next time you start Design Vision or reopen the GUI, it loads the preferences from this file.

For more information about GUI preferences, see the “Setting GUI Preferences” topic in online Help.

3

Performing Basic Tasks

In this chapter, experienced Design Compiler users who need to accomplish familiar basic tasks can find useful information about using Design Vision windows and menus to perform basic presynthesis and synthesis tasks. If you are new to Synopsys synthesis tools, see the *Design Compiler Tutorial Using Design Vision* to learn about basic tasks by performing them on a simple design. This chapter contains the following sections:

- Using Design Compiler Topographical Technology in Design Vision

Specifying Libraries

Before you start work on a design, specify the location of your libraries. You can define your library locations directly in the .synopsys_dc.setup file or indirectly by entering the locations in the Application Setup dialog box. Either method is acceptable—they both accomplish the same thing. (You can also specify library locations by running a script when you start Design Vision or by using the Execute Script dialog box. For details, see “Running Design Vision” on page 2-5 and “Using dctcl Scripts” on page 2-10.)

The link and target libraries are technology libraries that define the semiconductor vendor’s set of cells and related information, such as cell names, cell pin names, delay arcs, pin loading, design rules, and operating conditions. The symbol library defines the symbols for schematic viewing of the design.

You must specify any additional, specially licensed, DesignWare libraries with the `synthetic_library` variable (you do not need to specify the standard DesignWare library).

To define libraries in the .synopsys_dc.setup file, see the *Design Compiler User Guide*.

To specify the library location in the Application Setup dialog box,

1. Select File > Setup.

   The Application Setup dialog box opens.
2. Select the Defaults category if the Defaults page is not displayed.
3. Enter the appropriate path in the Search Path box.
4. Enter the library file names for the link, target, and symbol libraries you need to use.
5. (Optional) Enter the library file names for any specially licensed Synopsys or third-party DesignWare libraries you need to use.

   Note:
   > Synopsys provides a standard DesignWare library with components that implement many of the built-in HDL operators. You do not need to specify the standard DesignWare library.
6. Click OK.

For more information about the options on the Defaults page, see the “Setting Library Locations” topic in the online Help. For information about the options on the Variables page, see the “Setting Variables” topic.

To understand more about the function of link libraries, target libraries, symbol libraries, and DesignWare libraries, see the *Design Compiler User Guide*.

Reading In Your Design

To begin working on your design, read the design from disk into the tool's active memory. This is where all changes in the design take place before you save the design by writing it back to disk.

The File menu contains the commands for reading in a design:

- Analyze and Elaborate

  Use Analyze and Elaborate to read HDL designs and convert them to .ddc format. These commands open dialog boxes in which you can set options that are equivalent to the `analyze` and `elaborate` command-line options. For details, see the "Analyzing Files" and "Elaborating a Design" topics in online Help.

- Read

  Use Read (`read_file` is the command-line equivalent) to read designs that are already in.ddc format. This command opens a dialog box in which you can set options that are equivalent to the `read_file` command-line options. For details, see "Reading in a Design" topic in online Help.

The Analyze command checks the HDL designs for proper syntax and synthesizable logic, translates the design files into an intermediate format, and stores the intermediate files in the directory you specify.

The Elaborate command first checks the intermediate format files before building a.ddc design. During this process, Elaborate determines whether it has the necessary synthetic operators to replace the HDL operators, and it also determines correct bus size.

If you use Read to read in HDL files, the Analyze and Elaborate read functions are combined. However, Read does not perform certain design checks that Analyze and Elaborate perform.

Setting the Current Design

When you start a Design Vision session and read a design, the current design is automatically set to the top-level design. Some commands require you to set the current design to a subdesign before you issue them (the man pages provide such information).

To set the current design,

1. Click the drop-down list on the Design List toolbar to display the design names.
2. Select a design name.

Alternatively, you can open a design list view (by choosing List > Designs View), select a design name in the list, right-click, and choose Set Current Design. The command-line equivalent is `set current_design`.

Defining the Design Environment

Design Compiler requires that you model the environment of the design to be synthesized. This model comprises the external operating conditions (manufacturing process, temperature, and voltage), loads, drive characteristics, fanout loads, and wire loads. It directly influences design synthesis and optimization results.

Defining the design environment can involve using a large number of commands. Many designers find it convenient to define the design environment by using the default target technology library settings and by running scripts to define differences or additions. To define the design environment by using Design Vision menus, choose commands on the Operating Environment submenu under the Attributes menu. The Design Vision online Help has more information about particular commands and submenus under the Attributes menu.

Setting Design Constraints

Setting design constraints can involve using a large number of commands. Most designers find it convenient to use scripts to set design constraints.

Design Compiler uses design rule and optimization constraints to control the synthesis of the design.

Setting Design Rule Constraints

Design rules are provided in the vendor technology library to ensure that the product meets specifications and works as intended. Typical design rules constrain transition times, fanout loads, and capacitances. These rules specify technology requirements that you cannot violate. (You can, however, specify stricter constraints.)

To set design rule constraints for the current design by using the GUI,

- Choose Attributes > Optimization Constraints > Design Constraints, set options as needed, and click OK.

To set design rule constraints for certain input ports by using the GUI,

- Select the ports, choose Attributes > Optimization Directives > Input Port, set options as needed, and click OK.

Setting Optimization Constraints

Optimization constraints define the design goals for timing (clocks, clock skews, input delays, and output delays) and area (maximum area). During optimization, Design Compiler attempts to meet these goals; however, it does not violate your design rules. To optimize a design correctly, you must set realistic optimization constraints.

To set optimization constraints by using the GUI,

- Click Attributes in the menu bar to open the Attributes menu.

  Choose Specify Clock to set clock periods and waveforms. Other optimization constraints and settings are in the submenus under the Attributes menu: Operating Environment (input and output delays), Optimization Constraints (maximum and minimum delays and maximum area), and Optimization Directives (design attributes, object attributes, and timing exceptions). Explore these submenus to find the settings you need.

For more information about menu items in the Attributes menu, see the “Attributes Menu” topic in the online Help.

Defining Power Domains and Supply Networks

In multivoltage designs, the subdesign instances, or blocks, operate at different voltages. Power consumption can be reduced in a multivoltage design by making use of power domains that are independently powered up and down, including domains that are defined to have always-on relationships relative to each other.

By definition, a power domain is a logical grouping of one or more hierarchical blocks in the design. The blocks of a power domain can be powered up and down, independent of the power state of other power domains, except where a relative always-on relationship exists between two power domains. In particular, power domains can be defined and level-shifter and isolation cells can be used as needed to adjust voltage differences between power domains and to isolate shut-down power domains. For more details about power domains, see the *Power Compiler User Guide*.

In Design Vision, you can view information about the current power domains in your design and their power and ground supply networks. You can also define power domains, assign supply nets and supply ports, create and map power switches, and set and connect domain nets. For more details, see Design Vision online Help.

Compiling the Design

You can use Design Vision to initiate Design Compiler synthesis and optimization, thus compiling your high-level design description to your target technology.

Design Vision supports standard synthesis methodology: either a top-down compile or a bottom-up compile. For more information about compile methodologies, see the *Design Compiler User Guide*.

To compile the current design,

1. Choose Design > Compile Design.

   The Compile Design dialog box opens.

2. Select or deselect options as you require.

   For details about using the Compile Design dialog box, see the “Optimizing the Design” topic in online Help. Use the default settings for your first-pass compile. For most designs, the default settings provide good initial results. For more information about compile options, see the “Compile Design” topic in online Help, the *Design Compiler User Guide*, and the *Design Compiler Optimization Reference Manual*.

3. Click OK to begin compiling.

After compiling the design, save the design as described in the next “Saving the Design” on page 3-7.

The command log file records your Design Vision session. This log file is a record of all commands invoked. It includes commands you enter on the command line, commands initiated by your menu and dialog box selections, and tool commands (such as initialization commands and commands needed to execute user-entered commands).

You can use the command.log file to create a script file. Copy the file and use a text editor to add or remove commands as necessary.

Compiling High-Performance Designs

For high-performance designs that have significantly tight timing constraints, you can use Design Vision to initiate the DC Ultra solution for better quality of results (QoR). The Compile Ultra command is a push-button solution that allows you to apply the best possible set of timing-centric variables or commands during compile for critical delay optimization as well as improvement in area QoR.

Note:

> Because Compile Ultra includes all compile options and starts the entire compile process, no separate Compile command is necessary.

To compile the current design by using Compile Ultra,

1. Choose Design > Compile Ultra.

   The Compile Ultra dialog box opens.

2. Select or deselect options as you require.

   For details about using the Compile Ultra dialog box, see the "Optimizing Critical Delays" topic in online Help. Select options according to the requirements of your design. To perform a second-pass incremental compile, select the Incremental option. For more information about compile options, see the "Compile Ultra" topic in online Help, the *Design Compiler User Guide*, and the *Design Compiler Optimization Reference Manual*.

3. Click OK to begin compiling.

When you run Design Vision in topographical mode, the Compile Ultra command automatically uses the Design Compiler topographical features. All Compile Ultra command options are supported in this mode. Note that using the Incremental option with a topographical netlist results in placement-based optimization only. This compile should not be thought of as an incremental mapping. For more information about running Design Vision in topographical mode, see "Using Design Compiler Topographical Technology in Design Vision" on page 3-10.

To use the Compile Ultra command, you need a DC Ultra license and a DesignWare Foundation license.

Saving the Design

Design Vision does not automatically save designs before exiting.

To save the current design and each of its subdesigns in separate .ddc format files named *design_name*.ddc, where *design_name* is the name of the design,

- Choose File > Save.

To save the current design and all of its subdesigns in a single file with a different file name or file format,

- Choose File > Save As, enter or select a file name, select a file format, and click OK.

For more details, see the "Saving the Design" topic in online Help.

Working With Reports

Textual reports are available from the Design menu and the Timing menu. Use commands in the Design menu to generate design information and design object reports. Use commands in the Timing menu to generate timing and constraint reports. If these menus do not have the report you need, you can generate any Design Compiler report by issuing dc_shell report commands on the command line.

You can generate reports and display them in the report view (the default), save them in text files, or both. In a report view, you can select objects, save the report in a text file, or load a report from a file. In a report that contains object names (such as a design, cell, port, or net report), if you select an object name (blue text) in the report view, Design Vision cross-selects the object in a design schematic and magnifies the schematic to fit the object in the schematic view. For more information, see the “Viewing Reports” topic in online Help.

You can generate reports for selected objects.

To generate a report for a particular design object or group of design objects,

1. Select an object.
2. Choose a report command in the Design menu to open the associated dialog box.

   You can generate reports for cells, ports, and nets.
3. Click Selection in the report dialog box.

   The names of the selected objects appear.
4. Set other options as needed.
5. Click OK.

Printing Schematic and Symbol Views

You can print the path schematic or design schematic displayed in the active schematic view or the symbol diagram displayed in the active symbol view. Be sure that a default printer is set in your .cshrc file.

To print the active schematic or symbol view,

1. Generate a path schematic, design schematic, or symbol view.
2. Make sure the view you want to print is the active view.

   Click the corresponding tab at the bottom of the workspace if you need to make the view active.

3. Choose File > Print Schematic.

   The Select Printer Settings dialog box opens.

4. Select the print options you require and click OK.

You can also save an image of the schematic or symbol view in a PostScript file for printing later from a UNIX or Linux shell. For details about the options in the Print dialog box, see the “Printing Schematic and Symbol Views” topic in online Help.

Saving an Image of a Window or View

You can save an image of a top-level GUI window or view window in an image file. The image format can be PNG (the default), BMP, JPEG, or XPM. The image shows the window exactly as it appears on the screen but without the window border or title bar. For example, if you save an image of the active schematic view, the image shows the visible portion of the schematic at the current zoom level and pan position.

- To save an image of the current top-level window or the active view, use the Save Screenshot As dialog box.
- To save an image of any open GUI window or view window, use the `gui_write_window_image` command.

You cannot save images of dialog boxes or other GUI elements such as toolbars or panels

To save an image of the current window or active view,

1. Choose View > Save Screenshot As.

   The Save Screenshot As dialog box appears.

2. Select the file, or enter the path and file name in the "File name" box.

   The default format is PNG. you can specify a different format by using its extension to the file name.

3. (Optional) To save an image of the active view window instead of the top-level GUI window in which you are working, select the "Grab screenshot of active view only" option.
4. Click Save.

To save an image of any open GUI or view window, use the `gui_write_window_image` command to specify the file name, image format, and window name. Window instance names appear in the window title bars and on the Window menu.

Use the `-file` option to specify the file name. This option is required. For example, to save a PNG image of the active schematic view in a file named my_schematic.png, you can enter

```
design_vision> gui_write_window_image -file my_schematic
```

You can use a file name extension or the `-format` option to specify the image format. The default image format is PNG. For example, to save an XPM image of the active layout view in a file named my_layout.xpm, enter either of the following commands:

```
design_vision> gui_write_window_image -file my_layout.xpm
design_vision> gui_write_window_image -file my_layout -format xpm
```

Use the `-window` option to specify the window, For example, to save a PNG image of the layout window named Layout.1 in a file named mux_1.png, enter either of the following commands:

```
design_vision> gui_write_window_image -file mux_1 -window Layout.1
```

For more details about saving window and view images, see the “Saving an Image of a Window“ topic in online Help and the gui_write_window_image man page.

Using Design Compiler Topographical Technology in Design Vision

Design Compiler topographical technology provides the capability to accurately predict post-layout timing, area, and power during RTL synthesis without the need for wireload model-based timing approximations. It uses Synopsys’ placement and optimization technologies to drive accurate timing prediction within synthesis, ensuring better correlation to the final physical design. This new technology is a part of the DC Ultra feature set and is available only by using the `compile_ultra` command in topographical mode.

To use the Design Compiler topographical features in Design Vision, you must run Design Vision in topographical mode. In this mode, the command-line prompt appears as design_vision-topo in the console and the shell. For more details about starting Design Vision, see “Starting a Design Vision Session” on page 2-7.

Topographical technology leverages the Synopsys physical implementation solution to derive the “virtual layout” of the design so that the tool can accurately predict and use real net capacitances instead of wireload model-based statistical net approximations. If wireload models are present, they are ignored. In addition, the tool updates capacitances as synthesis progresses by adjusting placement-derived net delays based on an updated “virtual layout” at multiple points during synthesis.

This approach eliminates the need for over constraining the design or using optimistic wireload models in synthesis. The accurate prediction of net capacitances drives Design Compiler to generate a netlist that is optimized for all design goals including area, timing, test, and power. It also results in a better starting point for physical implementation.

Topographical technology supports all synthesis flows, including the following:

- Test-ready compile flow (basic scan and DFT MAX adaptive scan)
- Clock-gating flow
- Register retiming

When you use the Compile Ultra command in topographical mode, the Design Compiler topographical features are automatically used. All `compile_ultra` command options are supported. In addition, the Compile Design command (Design menu), the Report Wire Load command (Timing menu), and the Wire Load command (Attributes > Operating Environment menu) are not available in topographical mode.

For more information about using Design Compiler topographical technology, see the *Design Compiler User Guide*.

For information about working with Milkyway design libraries in topographical mode, see the following sections:

- Creating a Milkyway Design Library
- Opening and Closing a Milkyway Design Library
- Setting the TLUPlus Extraction Files

Creating a Milkyway Design Library

You need a Milkyway design library to save a design in Milkyway format for use in other Synopsys Galaxy platform tools, such as IC Compiler. You can use a single Milkyway design library across the entire Galaxy flow. Before creating a Milkyway design library in Design Vision, you must prepare the design and reference libraries. The Milkyway tool stores design data in the Milkyway design library and physical library data in the Milkyway reference library.

- The Milkyway directory structure used to store design data (the uniquified, mapped netlist and constraints) is referred to as the Milkyway design library. You can specify a design library for the current session by setting the `mw_design_library` variable to the root directory path.

- The Milkyway directory structure used to store physical library data is referred to as the Milkyway reference library. Reference libraries contain standard cells, macro cells, and pad cells. Design Vision uses the FRAM view of the reference libraries as the default physical model for your design. You can specify a reference library for the current session by setting the `mw_reference_library` variable to the root directory path.

When you create a Milkyway design library, Design Vision sets the reference libraries for the design. For more details about using Milkyway design libraries in topographical mode, see the *Design Compiler User Guide*.

To create a Milkyway design library,

1. Choose File > Create MW Library.

   The Create Library dialog box opens.
2. Specify the following design library information:
   - The path to the library root directory
   - The library name
   - The technology and physical library file names
3. Select a reference library option and specify the reference library information.

   You can specify the Milkyway reference library files directly or by using a reference control file. For details about using these options, see the “Creating a Milkyway Design Library” topic in online Help and the *Design Compiler User Guide*.
4. Set other options as needed.
5. Click OK.

Using the Create Library dialog box is equivalent to using the `create_mw_lib` command.

Opening and Closing a Milkyway Design Library

After you create a Milkyway design library, you must open it before you can read in the design. You cannot have more than one library open at the same time. If another library is already open, you must close it before opening a different library.

To open a Milkyway design library,

1. Choose File > Open MW Library.

   The Open Library dialog box opens.
2. Specify the library name and select a permission option.
3. Click OK.

Using the Open Library dialog box is equivalent to using the `open_mw_lib` command.

To close a Milkyway design library,

- Choose File > Close MW Library.

Using the Close MW Library command is equivalent to using the `close_mw_lib` command.

Setting the TLUPlus Extraction Files

TLUPlus is a binary table format in the Milkyway library for RC coefficients. Although you do not need to specify TLUPlus files if resistance and capacitance models are present in your vendor technology physical library, these files provide more accurate capacitance and resistance data, thereby improving correlation with back-end results.

To use TLUPlus, you must specify the maximum TLUPlus model files. You can also specify minimum TLUPlus model files and a map file that maps layer names between the Milkyway technology file and the process Interconnect Technology Format (ITF) file.

To select the TLUPlus files,

1. Choose File > Set TLU+.

   The Set TLU+ dialog box opens.
2. Specify the TLUPlus file names.
3. Click OK.

Using the Set TLU+ dialog box is equivalent to using the `set_tlu_plus_files` command.

Design Vision stores the names and locations of the TLUPlus files in Milkyway, but it does not store the TLUPlus information found in these files. For each Design Vision session you work on your design, you must specify the TLUPlus files that you need to use during the session.

4

Solving Timing Problems

This chapter presents procedures and suggestions for solving timing problems by using the features of Design Vision. The chapter does not provide details about exercising particular features of Design Vision, such as how to create a histogram or how to create a path schematic. For detailed information about Design Vision features, see the Design Vision online Help system.

This chapter contains the following sections:

Before You Analyze

Before you analyze your design with Design Vision, follow your normal compile methodology to create a constrained gate-level design. A constrained gate-level design is a prerequisite to any timing analysis.

For more information about using Design Vision to create a gate-level design, see Chapter 3, “Performing Basic Tasks.”

Creating a Timing Overview

Creating an overview of the timing of your design is a valuable way to start any analysis of your design's timing problems. A timing overview can help you decide what strategy to follow in gaining timing closure.

For example, a timing overview can help answer such questions as

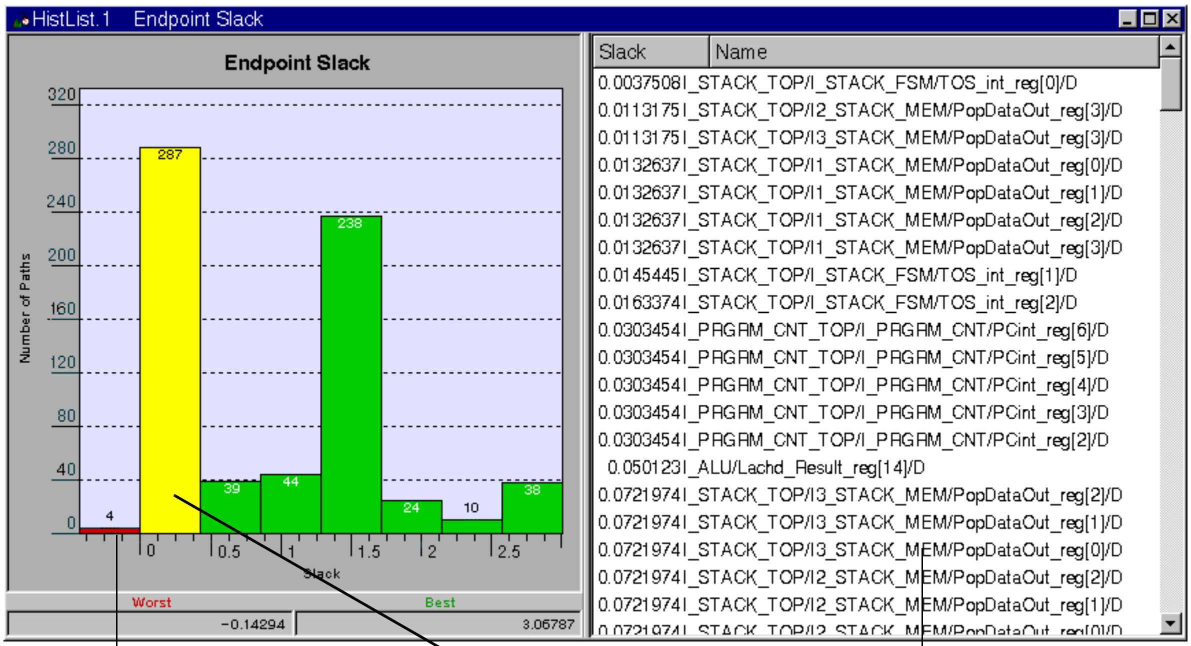
- Do I have many failing paths or just a few?
- Can I apply a local strategy for gaining timing closure?
- Do I need a global strategy for gaining timing closure?

To create a timing overview of your design,

1. Start with a constrained gate-level design.
2. Generate an endpoint slack histogram.

Figure 4-1 is a typical endpoint slack histogram for a design with a 4-ns clock cycle.

Figure 4-1 Endpoint Slack Histogram



Slack	Name
0.0037508	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[0]/D
0.0113175	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[3]/D
0.0113175	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[3]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[0]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[1]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[2]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[3]/D
0.0145445	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[1]/D
0.0163374	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[2]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[6]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[5]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[4]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[3]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[2]/D
0.050123	I_ALU/Lachd_Result_reg[14]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[0]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[0]/D

Endpoints of failing paths (red)

Selected bin (yellow)

Endpoint names for selected bin

Using information such as that in Figure 4-1, you might decide on a local strategy if just a few paths are failing by a small margin (failing path endpoints are in one or more red bins to the left of 0 on the horizontal axis). Conversely, if you find that many paths are failing, or that the design is failing your timing goals by a large margin, you might choose a higher-level, or global, strategy for problem solving.

Choosing a Strategy for Timing Closure

There is no single strategy that ensures quick and easy timing closure; however, a strategy based on the size and number of timing violations can be useful.

Assessing the Relative Size of Your Timing Violations

This section suggests guidelines for describing the relative size of timing violations in your design. After you create an endpoint slack histogram, you can use these size guidelines to help you judge what strategy to use for timing closure.

What you consider to be small or large violations depends on the requirements of your design and your design process; however, assessing violation size as a percentage of clock cycle can be useful.

- Small violations

  Some designers consider small violations to be about 10 percent of the clock cycle or less.

- Large violations

  Some designers consider large violations to be about 20 percent of the clock cycle or greater.

- Medium violations

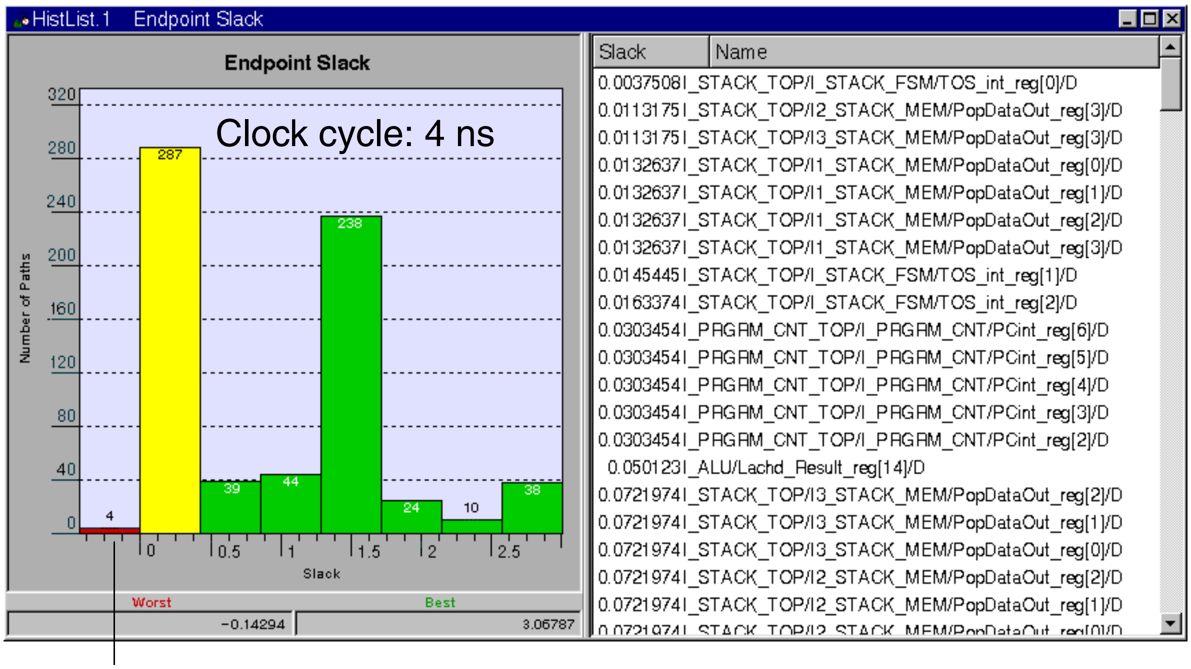
  Medium-size timing failures fall between the limits you set for large and small failures in your design or design process.

Whether your design is failing timing goals by large or small margins, the best strategy for timing closure is one that uses the least amount of runtime or number of design iterations to achieve timing goals. This principle underlies the methodology suggestions in this chapter. For more information about creating a timing overview, see “Creating a Timing Overview” on page 4-2.

When Timing Violations Are Small

Figure 4-2 is a histogram of a design that is failing timing goals by a small margin. For example, no path is failing by more than 0.14 ns—that is, less than 10 percent of the 4-ns clock cycle (ignoring input and output delay). You can click any bin to see the endpoint names for the paths in the bin.

Figure 4-2 Design With Small Timing Violations



Slack	Name
0.0037508	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[0]/D
0.0113175	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[3]/D
0.0113175	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[3]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[0]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[1]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[2]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[3]/D
0.0145445	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[1]/D
0.0163374	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[2]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[6]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[5]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[4]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[3]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[2]/D
0.050123	I_ALU/Lachd_Result_reg[14]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[0]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[0]/D

Endpoints of
failing paths (red)

Designs that fail by a small margin can have many failing paths or just a few. The endpoint slack histogram helps you to recognize quickly which case you have. Whether you have just a few failing paths or many, you can follow a global or local strategy in fixing the violations.

If suggestions for fixing small violations (either globally or locally) don't meet your timing goals, try applying the suggestions in "When Timing Violations Are Medium Size" on page 4-6 or "When Timing Violations Are Large" on page 4-8.

Working Globally to Fix Small Violations

To apply a global methodology for fixing small violations, consider recompiling your design using the incremental option and a higher map effort. The incremental option saves runtime by using the current netlist as the startpoint for design improvements.

The incremental compile with higher map effort has the advantage of simplicity—that is, it requires little or no time spent in analyzing the source of timing problems. However, this method can change much of the logic in the design.

Working Locally to Fix Small Violations

If you have a small number of paths with small violations, or if your violations seem to come from a limited set of problems on a few paths, a local strategy can be effective.

To use a local strategy for fixing small violations,

- Check hierarchy on failing paths

  Design Compiler does not optimize across hierarchical boundaries. Thus, snake paths limit Design Compiler's ability to solve timing problems on such paths.

- Look for excessive fanout on failing paths

  Because higher fanout causes higher transition times, excessive fanout can worsen negative slack on failing paths.

To check for hierarchy problems on failing paths,

1. Generate an endpoint slack histogram.
2. Click a bin that contains a failing path.

   A list of endpoints for failing paths is displayed.
3. Select the endpoint for the path you are interested in.
4. Generate a path schematic to see which leaf cells are in which levels of hierarchy.

   If your critical path, for example, crosses multiple subblocks of a level of hierarchy, consider ungrouping these subblocks. Design Compiler does not optimize across hierarchy boundaries. Thus, a subsequent compile has further opportunity to optimize the critical path when you ungroup such blocks.

To look for excessive fanout on failing paths,

1. Generate an endpoint slack histogram.
2. Select the endpoints for failing paths.

   Select the failing bin to see the endpoints.
3. Generate a timing report with the following options:
   - net
   - trans

Send the report output to the report view. For more information on report generation, see "Working With Reports" on page 3-8 and the "Viewing Reports" topic in online Help.

4. Examine the report for pins with high transition times and nets with high fanout.

   Such paths are candidates for buffering or drive-cell resizing.

5. Create path schematics of any paths you would like to see.

   A path schematic provides contextual information and details about the path and its components. Such information is often a prerequisite to understanding problems on the path.

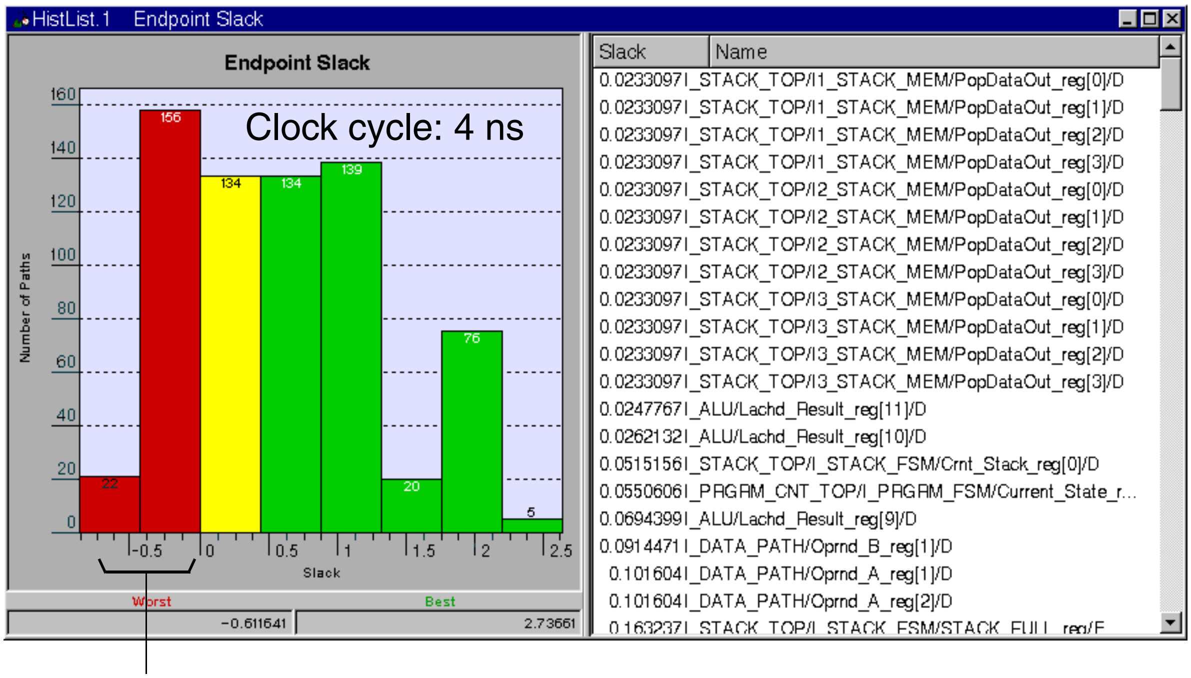
6. View fanin and fanout for path schematics.

   This step can provide useful information about the logic that drives, or is driven by, the problem path. For example, after viewing fanin or fanout, you might choose to resize cells in those logic cones.

When Timing Violations Are Medium Size

Figure 4-3 is a histogram of a design that is failing timing goals by margins that are between the large and small limits that are appropriate to your design methodology (for example, between 10 and 20 percent of the clock cycle). You can click a bin to see the endpoint names for paths the bin contains. A bin is yellow when selected. In Figure 4-3, one of the four bins containing endpoints of failing paths is selected.

Figure 4-3 Design With Medium Timing Violations



Slack	Name
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[0]/D
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[1]/D
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[2]/D
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[3]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[0]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[1]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[2]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[3]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[0]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[1]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[2]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[3]/D
0.0247767	I_ALU/Lachd_Result_reg[11]/D
0.0262132	I_ALU/Lachd_Result_reg[10]/D
0.0515156	I_STACK_TOP/I_STACK_FSM/Crnt_Stack_reg[0]/D
0.0550606	I_PRGRM_CNT_TOP/I_PRGRM_FSM/Current_State_r...
0.0694399	I_ALU/Lachd_Result_reg[9]/D
0.0914471	I_DATA_PATH/Oprnd_B_reg[1]/D
0.101604	I_DATA_PATH/Oprnd_A_reg[1]/D
0.101604	I_DATA_PATH/Oprnd_A_reg[2]/D
0.163237	I_STACK_TOP/I_STACK_FSM/STACK_FULL_reg/E

Endpoints of failing paths

When negative slack values are medium, you can use Design Vision to investigate further and focus your recompile on a critical range of negative slack values for path groups. Focusing your compile effort on a critical range can improve worst negative slack and total negative slack.

Defining a critical range for path groups offers the advantage of concentrating compile effort and runtime on those areas that most need it.

To investigate and focus a recompile by defining a critical negative slack range for path groups,

1. Create a path slack histogram for each path group in your design.

   Start with an arbitrary value of 1000 for the number of paths to include in each histogram. Raise or lower this value depending on the number of failing paths. The goal is to choose a value that shows you all or nearly all of the failing paths.

2. Decide on a critical range for each path group (note the values for use in step 3).

   When deciding on a critical range, choose a range that allows Design Compiler to focus on the worst endpoint violations without too large an increase in runtime. For example, some designers apply one of the following guidelines to decide on a critical range:

- Use a range that includes the worst 50 paths in a group.
- Use a range equal to one generic cell delay in your technology.

These are rough guidelines; for subsequent compiles you can adjust your critical range as necessary.

3. Set a critical range for each path group.

   Using the values you decided on in step 2, set the critical ranges for each path group with the `group_path` command. For example,

   ```
   design_vision> group_path -name my_clock -critical_range 0.25
   ```

4. Recompile the design.

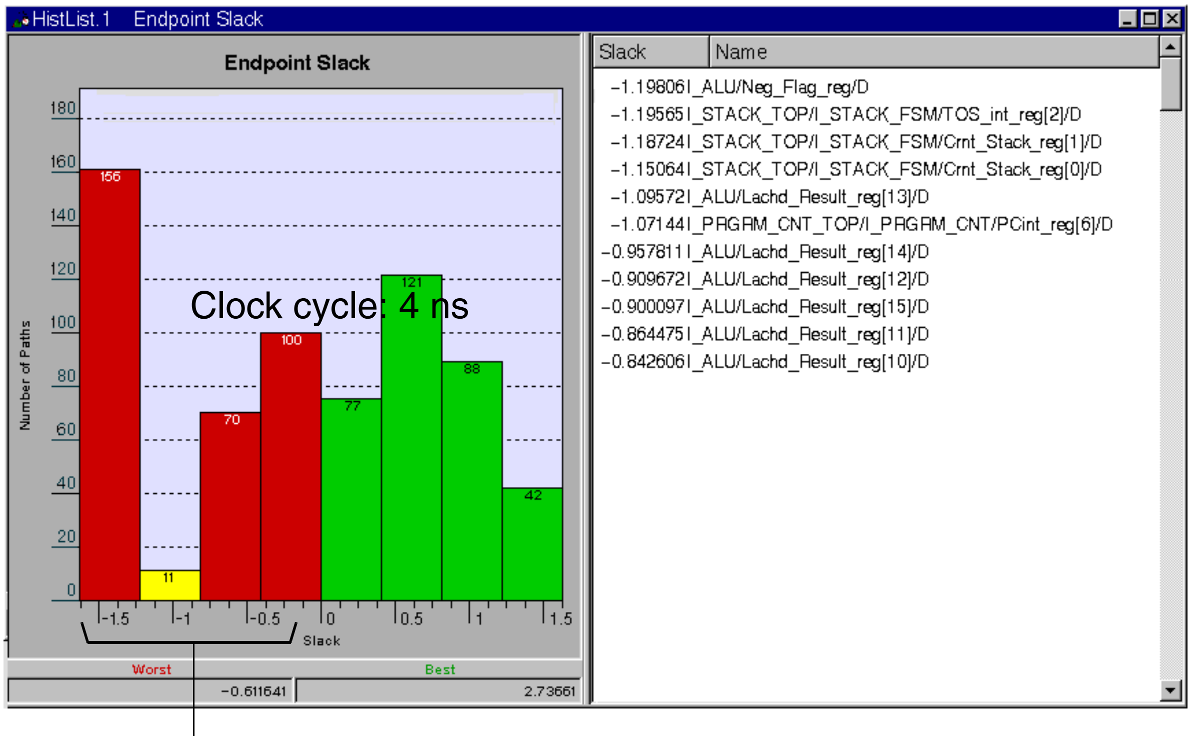
   With a critical range defined, the compile effort is now focused. However, you can also choose to increase the compile effort over your previous compile. In the Compile Design dialog box, select medium or high effort. Select the “Incremental mapping” option to direct Design Compiler to use the current netlist as a starting point for design improvements. An incremental compile can save runtime when your design does not require fundamental changes.

If suggestions in this section don’t meet your timing goals, try applying the suggestions in “When Timing Violations Are Large” on page 4-8.

When Timing Violations Are Large

Figure 4-4 shows a design that is failing timing goals by a large margin. You can click a bin to see the endpoint names for the paths it contains.

Figure 4-4 Design With Large Timing Violations



Slack	Name
-1.19806	I_ALU/Neg_Flag_reg/D
-1.19565	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[2]/D
-1.18724	I_STACK_TOP/I_STACK_FSM/Crnt_Stack_reg[1]/D
-1.15064	I_STACK_TOP/I_STACK_FSM/Crnt_Stack_reg[0]/D
-1.09572	I_ALU/Lachd_Result_reg[13]/D
-1.07144	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[6]/D
-0.957811	I_ALU/Lachd_Result_reg[14]/D
-0.909672	I_ALU/Lachd_Result_reg[12]/D
-0.900097	I_ALU/Lachd_Result_reg[15]/D
-0.864475	I_ALU/Lachd_Result_reg[11]/D
-0.842606	I_ALU/Lachd_Result_reg[10]/D

Endpoints of failing paths

Fixing large violations can require a high-level strategy to improve your design's performance. To fix large timing problems, consider any of the following changes:

- Modify your constraints.

  For example, increase the clock cycle or adjust time budgeting for the block or chip.

- Change the target technology.

  For example, target a higher performance technology.

- Modify the RTL.

  For example, you can move late-arriving signals such that you minimize their path length.

5

Solving Floorplan and Congestion Problems

The Design Compiler Graphical layout window provides tools that help you to analyze and debug physical problems related to Design Compiler topographical synthesis. In this chapter you can find general and specific information to help you

- Analyze the physical placement of critical timing path objects, where a cell might be placed at a physical distance from the rest of the path because of its fanin or fanout nets.
- Avoid correlation issues that can result from incorrect or missing physical constraints by visually examining the orientation and physical placement of objects such as macro cells, port locations, placement blockages, and the core area outline.
- Visually analyze floorplan-related congestion and identify the causes of congestion hotspots.

This chapter contains the following sections:

Physical View Advantage

As part of the Design Compiler Graphical package, Design Vision provides a layout window for viewing and analyzing the physical aspects of a design that you are optimizing by using the Design Compiler topographical technology. The layout window contains a layout view that displays physical design information such as

- Core placement area
- Ports
- Cells: hard and soft macros, I/O pads, ILM and physical hierarchy blocks, cell orientations, and selected or highlighted standard cells
- Pins
- Physical constraints: hard and soft placement blockages, site rows, bounds, preroutes, relative placement groups, voltage areas, and wiring keepouts
- Congestion map

The layout window is the physical design working environment for the GUI. Layout views provide the focal points for viewing and analyzing the physical layout of your design. The layout window provides the following tools:

- Visually customizable layout views
- An overview panel for quickly magnifying and traversing the active layout view or changing from one layout view to another
- A View Settings panel for controlling object visibility and selection and customizing object appearance in the active layout view
- A Map Mode panel for viewing the congestion map

Note that standard cells are not visible by default in the layout window. To view standard cells in the layout view, you must select or highlight them in another view, such as in a schematic view, or by choosing commands on the Select menu or Highlight menu. You cannot select or highlight standard cells by using tools in a layout view. Only a certain number of standard cells can be visible in the layout view at the same time.

Before You Start

Before you can view the physical layout of a design, you must

- Provide any necessary physical design setup information, such as libraries, TLUPlus files, preferred routing layer directions, and ignored layer settings.

For details, see the *Design Compiler User Guide*.

- Link the design without any errors.

You can view a design and analyze the physical aspects of a design in the layout window before or after you optimize the design. Before optimization, the layout window can display an elaborated GTECH design or a partially-synthesized design. You can use the layout window to

- Validate the physical constraints for your floorplan
- View the locations for ILM blocks, physical hierarchy blocks, and preplaced macro cells

  View cross-selected standard cells that have been mapped to specific locations by either the `set_cell_location` command or topographical technology virtual placement
- Select the cells in a logic design view such as the hierarchy browser or a schematic view

If you select unmapped GTECH cells or mapped standard cells that have not been assigned a location, they appear at the layout view origin (0,0). Note that bounds, relative placement groups, and the congestion map are not available in the layout window until you optimize the design.

To view a design after optimization, you must optimize the design using the Design Compiler topographical technology. You can either optimize the design during the current session or load the optimized design from a .ddc file.

After optimiziation, the layout window displays the optimized floorplan. You can

- Debug QoR Issues related to the physical aspects of your design, including
  - Why particular cells have a given drive strength
  - Why particular timing paths contain long buffer chains that are not related to high fanout
  - Why particular I/O paths contain high concentrations of buffers
  - What causes the huge transition or capacitance on particular pins
- Validate any user-defined physical constraints that you have applied to the design
- Analyze congested areas in the physical design

After performing QoR analysis, you can identify the next step, which might be one of the following:

- If there is a simple way to fix or eliminate the QoR issues, you might decide to continue with the back end flow.

- If you identify problems in the design source, such as your RTL, timing constraints, or physical constraints, you might need to rerun synthesis with updated source files.

The following steps illustrate the typical setup tasks in Design Vision before you can analyze a design in the layout window:

1. Start Design Vision in topographical mode:

   ```
   % design_vsion -topo -nogui
   ```

2. Set up the logic and physical technology libraries required for topographical mode.
3. Read in the .ddc netlist synthesized in topographical mode, and make sure that the design links correctly.
4. Open the GUI. Enter

   ```
   design_vision-topo> start_gui
   ```

5. Open the layout window.

Alternatively, you can perform steps 2 through 4 by running a dctcl script. The following example shows a basic setup script:

```
source echo dct.setup.tcl  <= sourcing DCT setup
read_ddc dct.opt.ddc  <= reading DCT synthesized DDC
current_design top
link
start_gui
```

Design Compiler Graphical features are enabled with the DC-Extension license, in addition to any other licenses for your current design configuration. These features are available in topographical shell (dc_shell-topo). If the DC-Extension license is not available, the tool issues the following error message:

```
Error: This site is not licensed for 'DC-Extension'. (SEC-51)
```

If you see this message, contact your local Synopsys representative.

Using the Layout Window

The Design Vision layout window has a similar user interface and the same look and feel as the IC Compiler layout window. However, the Design Vision layout window is designed to provide the features that you need to analyze and debug synthesis-related problems.

To open the layout window,

- In the Design Vision window, choose Windows > New Layout Window.

  The layout view, Overview panel, and View Settings panel are opened by default when you open the layout window.

Note:
When you open a layout window, all selected objects are deselected.

The layout view provides the focal point for viewing and analyzing your physical floorplan constraints and congested areas in your design. Use the Overview panel and the View Settings panel to adjust the layout view display when you examine objects and validate the applied constraints.

- You can magnify and traverse the layout view by clicking or dragging the pointer on the Overview panel. If multiple layout views are open, you can change the active view from one view to another.
- You can change layout view display properties by setting options on the View Settings panel, including object visibility, object selection, and object display styles.

The layout window is not designed to be used for the following applications:

- Floorplan exploration

  You cannot use the layout window as a floorplan exploration tool because it does not allow you to view user-supplied physical constraints until after you have performed topographical-based synthesis.
- Floorplan or physical constraint editing

  The layout window does not allow you to change any physical constraints by using the layout view. You must apply all required physical constraints before running the `compile_ultra` command. Design Vision does not support physical constraint changes between multipass synthesis runs.

If you significantly change or update the .ddc data, the layout window closes automatically.

To learn more about the layout window, see the Design Vision online Help.

Viewing the Physical Layout

The Design Vision layout view, like the IC Compiler layout view, displays a flat representation of the physical design, and it can display only one top-level design at a time. You do not need to set floorplan constraints before viewing a design in the layout view.

You can cross-select design objects between schematic and layout views. This helps you to understand the functions of the selected cells. When you select logic design objects in the hierarchy browser or a schematic view, the objects are automatically cross-selected in the layout view.

Layout data can be densely packed with overlapping objects. You can control the visibility (display or hide) and selection (enable or disable) of individual object types by setting options on the View Settings panel. You can also customize object properties such as color and fill pattern, and set other layout and object display options. If you open multiple layout views, you can set different options for each view.

You can open multiple layout views in a layout window. If you change settings in the active layout view and want to use the same settings in another layout view or during a future session, you can save them in your preferences file, .synopsys_dv_prefs.tcl. You can also restore previously saved display properties by loading them from your preferences file.

To open a layout view,

- Choose View > New Layout View.

You can visualize the physical layout as explained in the following sections:

- Navigating Through Layout Views
- Displaying Grid Lines
- Displaying Cell Orientations
- Drawing Rulers
- Examining ILM and Physical Hierarchy Blocks
- Setting and Saving View Properties

Navigating Through Layout Views

In the layout window, the Overview panel shows you what portion of the design is visible in each open layout view.

- The portion of the design displayed in the active layout view is shown as a solid yellow rectangle.
- The portions of the design displayed in other layout views are shown as solid gray rectangles.

You can use the Overview panel to quickly magnify or traverse the design in the active layout view. When multiple layout views are open, you can change to a different layout view. For more information about the Overview panel, see the Design Vision online Help.

Displaying Grid Lines

You can display or hide the lithography grid and the user grid. Both grids are hidden by default.

To display or hide the lithography grid,

- Choose View > Grid > Show Litho Grid.

To display or hide the user grid,

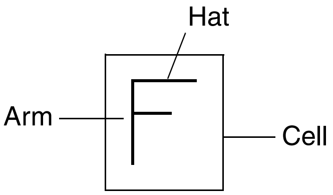
- Choose View > Grid > Show User Grid.

To switch between the default grid spacing and ten times the default grid spacing,

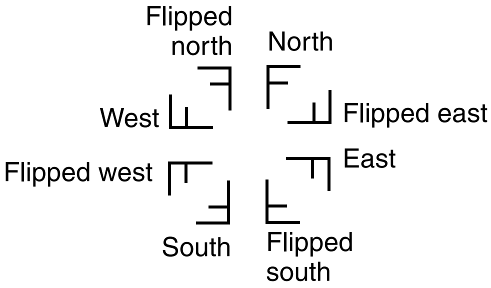
- Choose View > Grid > Cycle Grid Spacing.

Displaying Cell Orientations

Design Vision represents cell orientation in the layout view as an F. The position of the "hat" of the F indicates the first direction; the "arm" indicates the second direction.



Cells can be oriented in any of the following ways:



To display cell orientations in the active layout view,

1. On the View Settings panel, make sure the Cell visibility (Vis) option is selected.

2. Click the Cell expansion button (plus sign).
3. Select the Orientation visibility (Vis) option.

   A check mark on the option indicates that the cell orientations are visible.
4. Click Apply.

Drawing Rulers

You draw rulers to measure distances in a layout view. For example, you can measure the distance between two cells or between two pins on a net. A ruler can be composed of one or more horizontal, vertical, or diagonal segments. The distances from the beginning of the ruler are labeled at the ends of each segment and at every tenth tick mark within a segment.

While you draw a ruler segment, the GUI displays a preview of the ruler in the layout view and displays the coordinates for the current pointer position on the status bar. The preview image indicates the distance between the initial point and the current pointer position.

To draw a ruler,

1. Click the Ruler tool button on the Mouse Tools toolbar or choose View > Mouse Tools > Ruler Tool.
2. Click the location in the layout view where you want to begin the ruler.
3. Move the pointer in the direction that you want to draw the ruler segment, and click to define the segment.

   To draw a diagonal segment, press the Shift key when you move the pointer.
4. Click where you want to end the ruler segment.

   The ruler segment appears.
5. Repeat steps 2 through 4 to draw additional ruler segments.
6. When you finish the last segment, press the Escape key or right-click and choose End Ruler.

Examining ILM and Physical Hierarchy Blocks

If your design contains ILM or physical hierarchy blocks, you should visually check the blocks for correct site locations when you validate the floorplan. You should also visually check for ILM blocks in highly congested areas when you examine the congestion map.

An ILM is a structural model of a circuit that is modeled as a smaller circuit representing the interface logic of the block. In an ILM block, the gate-level netlist for the block is modeled by another gate-level netlist that contains only the interface logic of the block. ILMs contain cells whose timing affects or is affected by the external environment of a block.

A physical hierarchy is a block that you create from a hierarchical cell by using the `set_physical_hierarchy` command. For details about this command, see the man page. The layout view displays a physical hierarchy as a rectangular block that indicates the area in which the leaf cells have been placed. You can view cell placement, pin placement, or net connections within the block by selecting the cells, pins, or nets in the hierarchy browser or a schematic view. The selected objects are cross-selected in the layout view and displayed in the selection color, which is white by default.

You can set options on the View Settings panel to control the visibility and selection of ILM and physical hierarchy blocks in the layout view and to change their color and fill pattern. You can distinguish ILM blocks from physical hierarchy blocks by the value of the `cell_type` attribute, which you can view by using the Query tool or the Properties dialog box.

- The cell type for an ILM block is ILM.
- The cell type for a physical hierarchy block is Physical Hierarchy.

You can expand ILM blocks to display the objects (cells, ports, pins, nets, and so forth) inside the blocks. When ILM blocks are expanded, you can use layout window analysis and editing tools to analyze and modify the placement of cells within the blocks. You can control the visibility and selection of objects within ILM blocks by using the same View Settings panel options you use for other objects in the layout view.

To expand ILM blocks,

1. Select a value greater than 0 in the “View level” option on the View Settings panel.
2. Click Apply.

To collapse (close) ILM blocks,

1. Select 0 in the “View level” option on the View Settings panel.
2. Click Apply.

For more information about viewing ILM and physical hierarchy blocks, see the Design Vision online Help. For more information about the Design Compiler hierarchical flow, see the *Design Compiler User Guide*.

Setting and Saving View Properties

The View Settings panel provides options you can use to set display properties in the active layout view. You can also save the current settings in your preferences file, or load settings from the preferences file. If you open multiple layout views, you can set different options for each view.

To display or hide the View Settings panel,

- Choose View > Toolbars > View Settings.

  A check mark beside the command on the Toolbars menu indicates that the View Settings panel is visible.

You can set options on the View Settings panel to

- Control object visibility and selection
- Display or hide object labels
- Change object display styles such as colors or fill patterns
- Set layout view display options for cell orientations (display or hide), the brightness level, and the hierarchy level for ILM blocks.

You can set visibility or selection options or change style properties for object types or subtypes. Object subtypes are categories of objects by property or attribute. For example, when cells are visible, you can display core cells and hide macro cells.

By displaying or hiding particular object types, you can visually inspect just the physical layout data that you are interested in viewing while ignoring unrelated data. By enabling or disabling the selection of particular object types, you can control which types of objects are selected when you click or drag the pointer in a layout view.

To change layout view display properties in the active layout view,

1. Set options as needed on the View Settings panel.
2. Click Apply.

By default, when you change settings on the View Settings panel, you must click Apply before the changes take effect in the active view. If you prefer, you can set the panel to automatically apply your changes as soon as you make them.

To enable or disable the automatic apply mechanism,

- Choose Options > Auto Apply.

A check mark beside the command on the Options menu indicates that the auto apply mechanism is enabled.

Alternatively, when the automatic apply mechanism is not enabled, you can reverse changes that you have not already applied.

To reverse unapplied changes,

- Choose Options > Cancel Changes.

You can customize how objects appear in the layout view by changing their style properties. Object styles set the appearance of objects in the active layout view. You can set the color, fill pattern, outline style, outline width, or exaggeration value for individual object types or layers.

Design Vision does not save view settings when you exit the session. If you change layout view settings during a session and want to use the same settings in a future session, you can save them in your preferences file. You can also restore previously saved view settings by loading them from your preferences file.

To save the current settings for the active layout view,

- Choose Options > Preferences > Save to Preferences on the View Settings panel.

To restore the most recently saved layout view settings,

- Choose Options > Preferences > Set from Preferences on the View Settings panel.

For more information about the View Settings panel, see the Design Vision online Help.

Validating Physical Constraints

The layout view provides visual feedback about the orientation and physical placement of physical design objects. By visually examining these objects, you can avoid the correlation problems that can occur due to incorrect or missing physical constraints. You can examine the physical constraints in your floorplan by viewing and probing them in the layout view in the following ways:

- View the core area, site rows, bounds, placement blockages, preroutes, relative placement groups, voltage areas, and wiring keepouts
- Select or highlight the core area, bounds, placement blockages, relative placement groups, and wiring keepouts
- Query (display information about) the core area, bounds, placement blockages, preroutes, relative placement groups, voltage areas, and wiring keepouts

In addition, you can view object properties for the core area, bounds, placement blockages, relative placement groups, and wiring keepouts by using the Properties dialog box (choose Edit > Properties).

Physical constraint validation provides the following benefits:

- Helps you improve the physical constraints and achieve better results

  For example, you can identify the need for placement blockages to plug gaps between macros that the synthesis tool might consider free to use but your physical implementation tool does not use.

- Helps you identify mismatched results between Design Compiler topographical technology and IC Compiler.

  For example, incorrect application of physical constraints during synthesis can lead to ignored placement blockages in the physical implementation tool.

Note:

> Standard cells are not visible by default in the layout view. You must select or highlight standard cells in the hierarchy browser or a schematic view, or by choosing commands on the Select menu, to view them in the layout view.

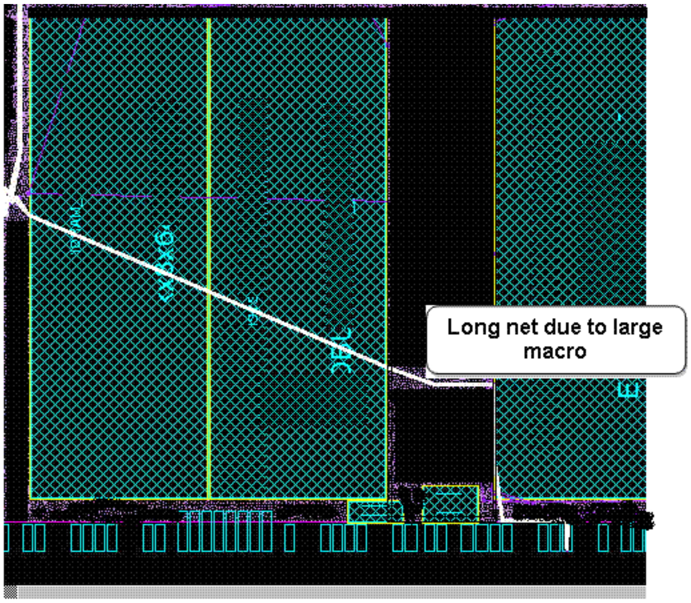
Debugging QoR Issues Related to the Floorplan and Placement

The Design Vision layout view allows you to debug the physical design problems that cause QoR degradation, especially timing degradation. The layout view provides visual feedback about the physical placement of timing path objects. By visually examining the critical path in the layout view, you can find answers to such questions as

- Why are certain cells of a given drive strength?
- Why does a path contain long buffer chains that are not related to high fanout?
- Why are certain cells placed at a physical distance from the rest?
- Why are there high concentrations of buffers on I/O path?
- Why is there high transition or capacitance on pins?

You can query and highlight design objects on the critical path to find answers to these questions that help you understand Design Compiler topographical placement and the problems that can cause timing degradation.

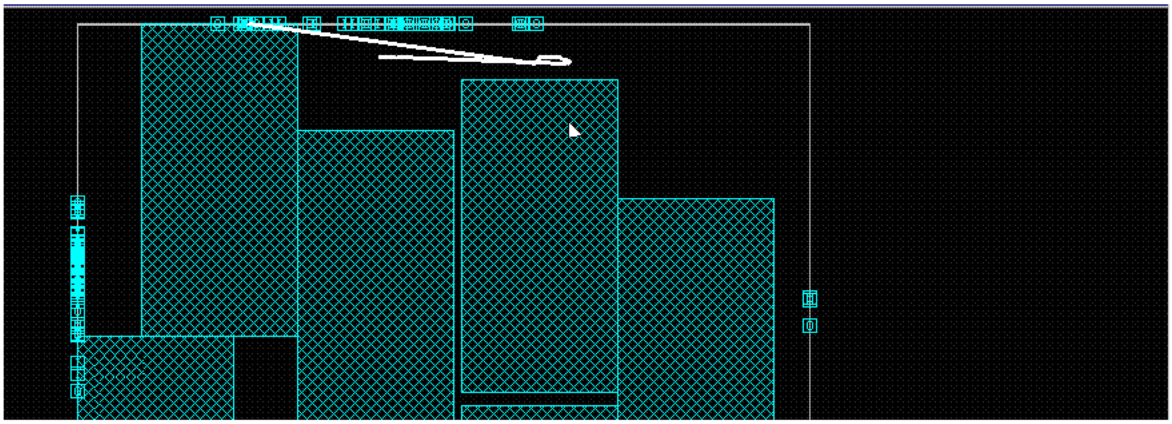
For example, you can perform critical path analysis in the layout view to identify the kinds of physical problems that can cause QoR degradation, such as why a long net is on the critical path.



To view the critical path,

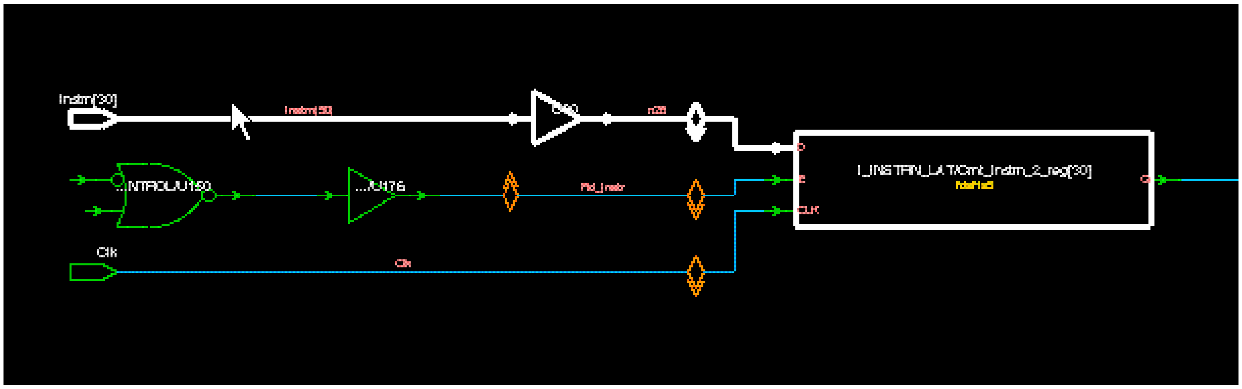
1. In the Design Vision main window, choose Timing > Timing Analysis Driver.
2. In the timing analysis driver, select the path with the worst negative slack.

The selected path is automatically cross-selected in the layout view.

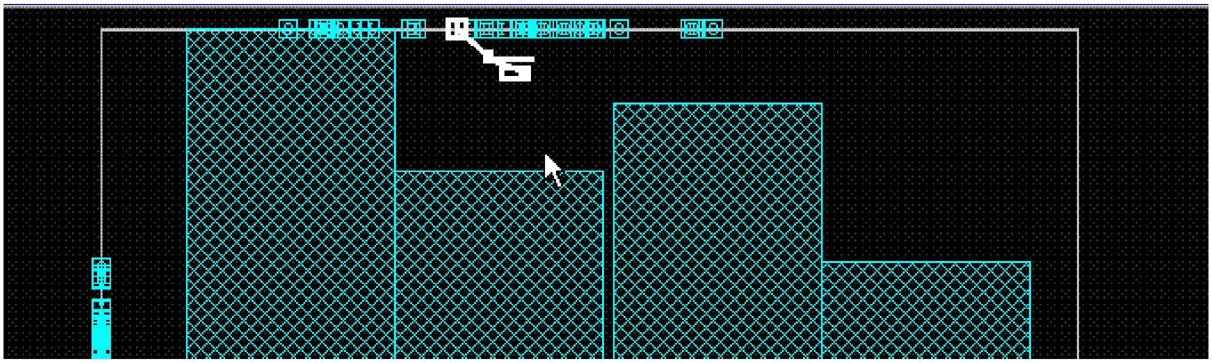


You can analyze the critical path to understand the Design Compiler topographical placement. The following example demonstrates one way to identify why the path startpoint is at a particular location:

1. Select the path startpoint in a path schematic.
2. Right-click and choose Add Logic > Fanin/Fanout to open the Select Fanin/Fanout dialog box.
3. Select the Fanin option, set other options as needed, and click OK.
4. Select the startpoint and its input path.



When you select the startpoint and input path in the path schematic, the selected logic is automatically cross-selected in the layout view.

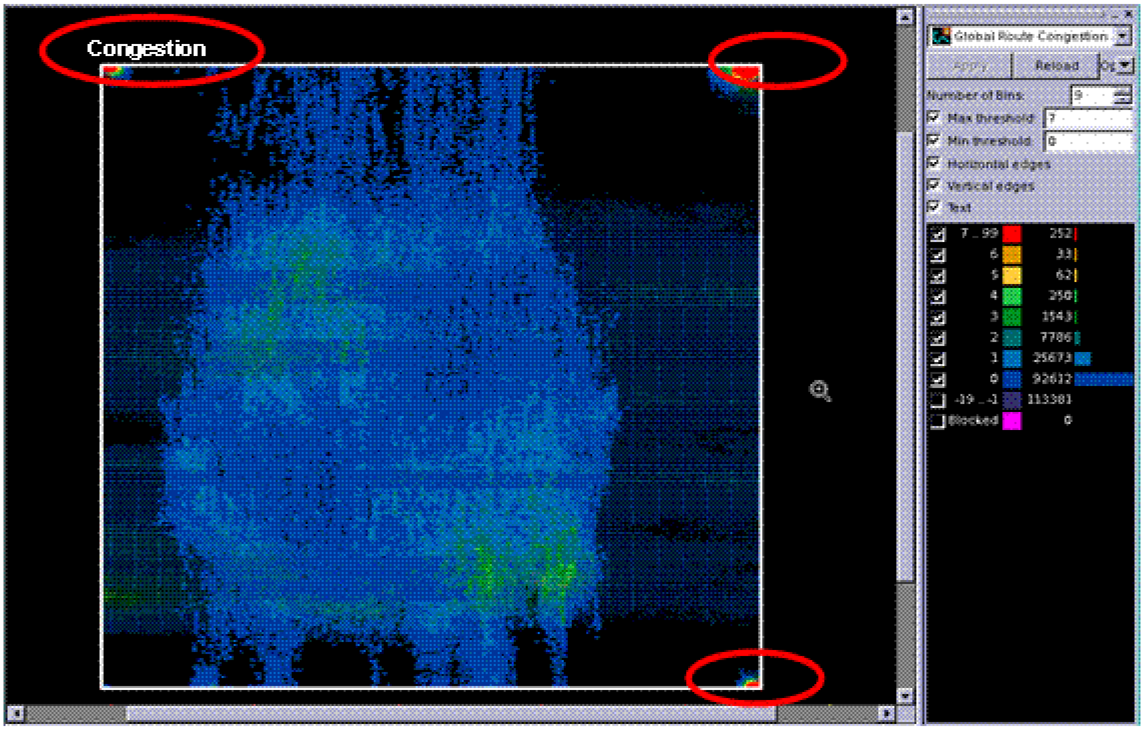


Visually Analyzing Congestion

You can identify areas of high congestion in your design by viewing the congestion map. By visually examining congested areas in your design, you can determine whether the design is routable and identify the causes of the congestion if the design is not routable. You can display or hide the congestion map at any time when a layout view is open in the layout window.

Figure 5-1 shows an example of congestion resulting from the layout of the floorplan.

Figure 5-1 Floorplan Congestion in Layout View



Number of Bins: 9
☑ Max threshold: 7
☑ Min threshold: 0
☑ Horizontal edges
☑ Vertical edges
☑ Text

☑	7 .. 99	🟥	252
☑	6	🟧	33
☑	5	🟨	62
☑	4	🟩	250
☑	3	🟩	1543
☑	2	🟦	7786
☑	1	🟦	25673
☑	0	🟦	92612
□	-19 .. -1	🟪	113381
□	Blocked	🟪	0

To display or hide the congestion map,

- In the layout window, choose View > Map Mode.

The GUI dims the visible objects in the layout view and displays the Map Mode panel. If you have already generated congestion data during the session, the congestion map grid appears on top of the design in the layout view. If the map does not appear, you must load the congestion data. You can reload the data if it changes during the session.

To load or reload the congestion data,

1. Click the Reload button on the Map Mode panel.
2. Choose View > Map Mode.

The congestion map divides the core area into a grid of colored boxes. Each box represents a vertical plane and a horizontal plane through which routes can pass. The left and bottom box edges are colored and labeled to show the usage-to-capacity ratios of routing tracks through the planes. Each map color represents a range of congestion values called a bin. The ranges are calculated by using a linear interpolation of the congestion data between minimum and maximum thresholds.

For information about viewing the congestion in the layout view, see the following sections:

- Viewing the Congestion Map
- Viewing Cells in Congested Areas

For more in depth information about congestion in the Design Vision Graphical, see the *Design Compiler User Guide*.

Viewing the Congestion Map

You can view map information in the legend on the Map Mode panel, and you can display or hide individual map colors (bins). The legend displays the color, the data count, and optionally, the color exaggeration level for each bin. The colored histogram bars on the right side of the legend represent the relative distribution of congestion values in the bins. You can use the visibility options on the left side of the legend to display or hide map colors for individual bins.

To display or hide map colors in the layout view,

1. Select the visibility options for the colors you want to display, and deselect the visibility options for the colors you want to hide.
2. Click Apply.

You can facilitate your congestion analysis by

- Displaying or hiding map details (box edges, labels, or layers)
- Recalculating the congestion ranges

To display or hide map details,

1. Set the map display options as needed.
   - To display or hide the horizontal or vertical box edges, select or deselect the “Horizontal edges” or “Vertical edges” option.
   - To display or hide congestion labels, select or deselect the Text option.
   - To display or hide individual layers, select or deselect the options for those layers.
2. Click Apply.

To adjust the congestion ranges,

1. Set the congestion thresholds as needed.
   - To change the congestion thresholds, type values in the “Max threshold” and “Min threshold” boxes.
   - To apply or remove the congestion thresholds, select or deselect the “Max threshold” and “Min threshold” options.
   - To change the number of congestion ranges (bins), select or type a value in the Bins box.
2. Click Apply.

To learn more about the congestion map and the Map Mode panel, see Design Vision online Help.

Viewing Cells in Congested Areas

When the congestion map is enabled, you can view and select cells in congested areas of the design.

Use the “List cells in congested region” dialog box to view and select cells in congested areas of the design.

To select and view cells in a congested region,

1. Click the “List cells in congested region” button on the Map Mode panel.

   The “List by Congested Region” dialog box appears.

   You can move this dialog box to a location on the screen where you can work with both it and the layout view at the same time.

2. Define the shape and location for the region by doing one of the following:
   - Drag the pointer in the layout view to form the rectangle where you need it.
   - Click the button ⬚ and type the x- and y- coordinates for the upper left and lower right corners of the rectangle in the Coordinates box.
3. Click Apply.

   The names of the cells in the region appear in the Cell Name list. Only cells in highly congested areas are listed.
4. Select cells in the list that you want to view in the layout view.

   The cells appear in the select color (white by default) in the layout view.

Index